2024-1971

In The

United States Court Of Appeals
For The Federal Circuit

In re: WAG ACQUISITION, LLC,

Appellant.

ON APPEAL FROM THE UNITED STATES PATENT AND TRADEMARK OFFICE
PTO-1 : 90/014,833

JOINT APPENDIX

Ronald Abramson
Ari J. Jaffess
LISTON ABRAMSON LLP
The Chrysler Building
405 Lexington Avenue, 46th Floor
New York, New York 10174

Counsel for Appellant

Brian Racilla
Monica B. Lateef
Amy J. Nelson
UNITED STATES PATENT AND TRADEMARK OFFICE
Office of the Solicitor
PO Box 1450
Mail Stop 8
Alexandria, Virginia 22313

Counsel for Appellee

APPENDIX TABLE OF CONTENTS*

In re: WAG ACQUISITION, LLC
No. 2024-1971

Materials Required Pursuant to Fed. Cir. R. 30.

* The leading "Exhibit" references in the section titled "Chronological Record in Reexamination Control No. 90/014,833" refer to designations from the exhibits of the Request for *Ex Parte* Reexamination. None of the remaining documents included in the Joint Appendix have an identifying designation in the reviewing tribunal. All remaining documents are either from the file history of the reexamination proceeding (which does not include an identifying designation for its documents), or are other documents, such as the Certified List (required by Fed. Cir. R. 17(b)(1)).

8/25/2021	Exhibit 15, Plaintiff's Opening Claim Construction Brief, No. 14-cv-02832 (D.N.J.)	Appx1851-1852, Appx1876-1879
8/25/2021	Exhibit 16, Plaintiff's Responsive Claim Construction Brief, No. 14-cv-02832 (D.N.J.)	Appx1892-1893, Appx1915-1918
8/25/2021	Exhibit 17, Plaintiff's Opening Claim Construction Brief, No. 19-cv-01278 (W.D. Wa.)	Appx1929-1930, Appx1953-1956
8/25/2021	Exhibit 18, Plaintiff's Responsive Claim Construction Brief, No. 19-cv-01278 (W.D. Wa.)	Appx1962-1963, Appx1972-1975
11/16/2021	Order Granting Request for *Ex Parte* Reexamination	Appx2733-2735
3/4/2022	Office Action in *Ex Parte* Reexamination	Appx2774-2775, Appx2778-2779, Appx2802-2803
6/6/2022	Response to Office Action	Appx2928-2930
7/27/2022	Final Rejection Cover Sheet	Appx2958-2958
7/27/2022	Final Rejection	Appx2959-3064
9/28/2022	Notice of Appeal to PTAB	Appx3072-3073
11/28/2022	Patent Owner's Appeal Brief Before PTAB	Appx3090, Appx3104-3106, Appx3118-3120, Appx3127-3130

3/15/2023	Examiner's Answer Before PTAB	Appx3620-3627, Appx3636-3638, Appx3643-3650
5/15/2023	Patent Owner's Reply Brief Before PTAB	Appx3676, Appx3700-3703
9/18/2023	Record of Oral Hearing, *Ex Parte WAG ACQUISITION*, *LLC*, Appeal Nos. 2023-002525, 2023-003028, 2023-003319 (P.T.A.B.)	Appx3748-3750, Appx3769-3771
1/16/2024	Request for Rehearing Before PTAB	Appx3827-3838
6/13/2024	Patent Owner's Notice of Appeal to Federal Circuit	Appx3884-3907

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/014,833	08/25/2021	8327011	125737.538655	1250

7590 11/17/2023

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES, LLC
231 SOMERVILLE ROAD
BEDMINSTER, NJ 07921

EXAMINER	
WASSUM, LUKE S	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
11/17/2023	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte WAG ACQUISITION, LLC
Patent Owner and Appellant

Appeal 2023-002525
Reexamination Control 90/014,833
Patent 8,327,011 B2
Technology Center 3900

Before JOHN A. JEFFERY, ERIC B. CHEN, and BRIAN J. McNAMARA, *Administrative Patent Judges*.

CHEN, *Administrative Patent Judge*.

DECISION ON APPEAL

Pursuant to 35 U.S.C. §§ 134(b) and 306, Patent Owner[1] appeals from the final rejection of claims 1 and 4. Claims 2 and 3 are not subject to reexamination.

A video oral hearing was held on September 18, 2023. The record includes a written transcript of the oral hearing. We have jurisdiction under 35 U.S.C. § 6(b).

We AFFIRM.

---

[1] Patent Owner identifies the real party in interest as WAG Acquisition, LLC. (Appeal Br. 1.)

STATEMENT OF THE CASE

Reexamination Proceedings

A request for *ex parte* reexamination of U.S. Patent No. 8,327,011 B2 ("the '011 patent") was filed on August 25, 2021 and assigned Control No. 90/014,833. The '011 patent, entitled "Streaming Media Buffering System" issued December 4, 2012 to Harold Edward Price, based on Application No. 13/374,942, filed January 24, 2012. The '011 patent is part of a series of multiple continuing applications, with the first application filed on March 28, 2001, which claims priority to provisional Application No. 60/231,997, filed on September 12, 2000.

Claimed Subject Matter

The claims are directed to streaming media, sent via the Internet, such that the media in the user buffer accumulates and immediately played on a user's computer. (Abstract.)

Related Litigation

The '011 patent has been asserted in the following litigation: (i) *WAG Acquisition, LLC v. Sobonito Investments, Ltd. et al.*, No. 2:14-cv-1661 (D.N.J.) (dismissed); (ii) *WAG Acquisition, LLC v. Multi Media, LLC et al.*, No. 2:14-cv-2340 (D.N.J.) (dismissed); (iii) *WAG Acquisition, LLC v. Data Conversions, Inc. et al.*, No. 2:14-cv-2345 (D.N.J.) (dismissed); (iv) *WAG Acquisition, LLC v. Flying Crocodile, Inc. et al.*, No. 2:14-cv-2674 (D.N.J.); (v) *WAG Acquisition, LLC v. Gattyán Group S.á.r.l. et al.*, No. 2:14-cv-2832 (D.N.J.) (dismissed); (vi) *WAG Acquisition, LLC v. FriendFinder Networks Inc., et al.*, No. 2:14-cv-3456 (D.N.J.); (vii) *WAG Acquisition,*

LLC v. Vubeology, Inc. et al., No. 2:14-cv-4531 (D.N.J.) (dismissed); (viii) *WAG Acquisition, LLC v. Gamelink Int'l Ltd. et al.*, No. 2:15-cv-3416 (D.N.J.) (dismissed); and (ix) *WAG Acquisition LLC v. WebPower, Inc. et al.*, No. 2:15-cv-3581 (D.N.J.) (dismissed).

The '011 patent was subject to the following petitions for *inter partes* reviews: (i) *FriendFinder Networks Inc. et al. v. WAG Acquisition LLC*, IPR2015-01033 (PTAB Oct. 19, 2015) (institution denied); (ii) *I.M.L. SLU v. WAG Acquisition LLC*, IPR2016-01655 (PTAB Feb. 27, 2017) (institution denied); and (iii) *WebPower v. WAG Acquisition LLC*, IPR2016-01161 (PTAB Dec. 12, 2016) (institution denied).

The Claims

Claims 1 and 4 are illustrative of the claimed subject matter, and reproduced below with disputed limitations in italics:

> 1. A media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number, comprising
>
> a processor;
>
> a memory;
>
> a connection to the network; and
>
> media player software comprising
>
> instructions to cause the media player to request from the media source a predetermined number of media data elements;
>
> instructions to cause the media player to receive media data elements sent to the media player by the

media source and store the media data elements in the memory;

instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to *maintain a record of the serial number of the last media data element that has been received and stored in the player buffer*;

instructions to cause the media player to play media data elements sequentially from the player buffer; and

instructions to cause the media player to *transmit to the media source a request to send one or more media data elements, each identified by a serial number*, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.

4. The media player of claim 1, *wherein the instructions* for causing the media player to request from the media source a predetermined number of media data elements *further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player.*

REFERENCES

Name	Reference	Date
Glaser et al.	US 5,793,980	Aug. 11, 1998
Imai et al.	US 5,987,510	Nov. 16, 1999
Hill	US 6,005,600	Dec. 21, 1999
Carmel et al.	US 6,389,473 B1	May 14, 2002
Shteyn	US 7,529,806 B1	May 5, 2009

REJECTIONS

A. Claims 1 and 4 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Shteyn.

B. Claim 4 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Shteyn and Carmel.

C. Claim 4 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Shteyn and Glaser.

D. Claims 1 and 4 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Hill.

E. Claim 4 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Hill and Carmel.

F. Claim 4 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Hill and Glaser.

G. Claims 1 and 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Shteyn and Hill.

H. Claims 1 and 4 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Carmel.

I. Claims 1 and 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Carmel and Hill.

J. Claim 1 stands rejected under 35 U.S.C. § 102(e) as being anticipated by Imai.

K. Claims 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Imai and Carmel.

L. Claims 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Imai and Glaser.

OPINION

§ 102 Rejection—Hill

<u>Independent Claim 1</u>

We are unpersuaded by Patent Owner's arguments (Appeal Br. 34–35; *see also* Reply Br. 22–23) that Hill does not describe the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number," as recited in independent claim 1.

The Examiner found that buffer manager 118 of Hill, which retrieves files from source 122, corresponds to the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number." (Final Act. 15–16; *see also* Ans. 25–26.) We agree with the Examiner's findings.

Hill relates "to playing movie data held in various storage devices with varying bandwidth capabilities, where the movie player adapts to these varying conditions." (Col. 1, ll. 7–10.) Figure 1 of Hill, reproduced below, illustrates digital network environment 102. (Col. 3, ll. 27–28.)

FIG. 1

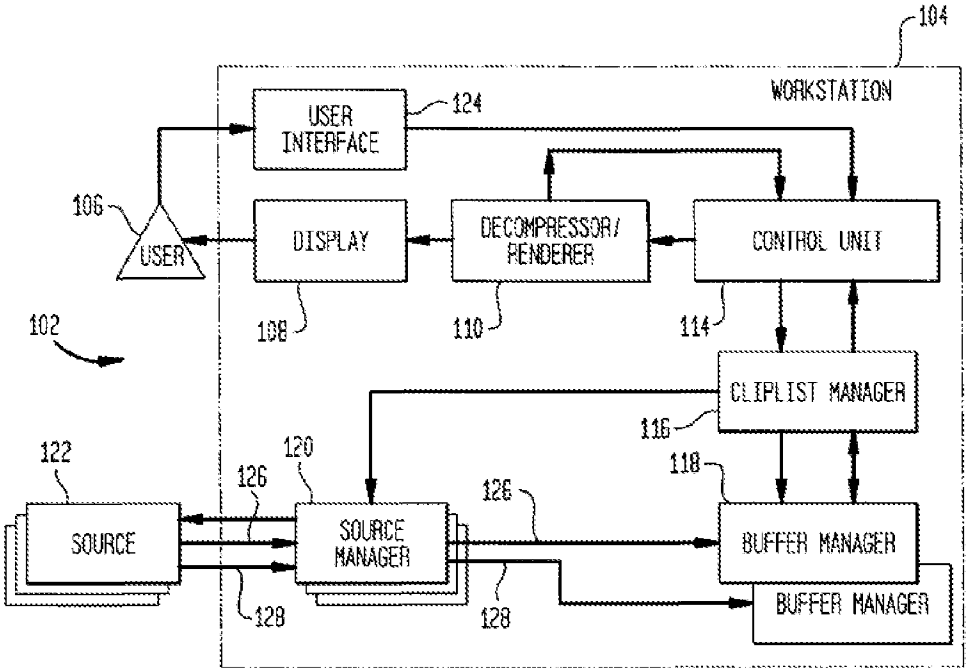


Figure 1 illustrates digital network environment 102.

In reference to Figure 1, Hill explains that digital network environment 102 features "[a] user 106 [that] utilizes the workstation 104 to play movie data." (Col. 3, ll. 27–30.) Hill further explains that "the movie data of interest to the user contains N_c clips, each stored in a separate source 122," such that "[e]ach source 122 may be any type of digital memory, and may be accessed either locally or via a network server" or "a source may be a site on the Internet." (Col. 3, ll. 49–56.) Additionally, Hill also explains that "[t]he buffer manager 118 provides frames upon request to the control unit 114 via the cliplist manager 116" (col. 4, ll. 7–8) and "[t]he cliplist manager 116 requests frames from a particular source 122 via the appropriate source manager 120" (col. 4, ll. 27–28).

Figure 3 of Hill, reproduced below, illustrates flowchart 320 of the input/output operation for buffer manager 18. (Col. 2, ll. 64–65.)

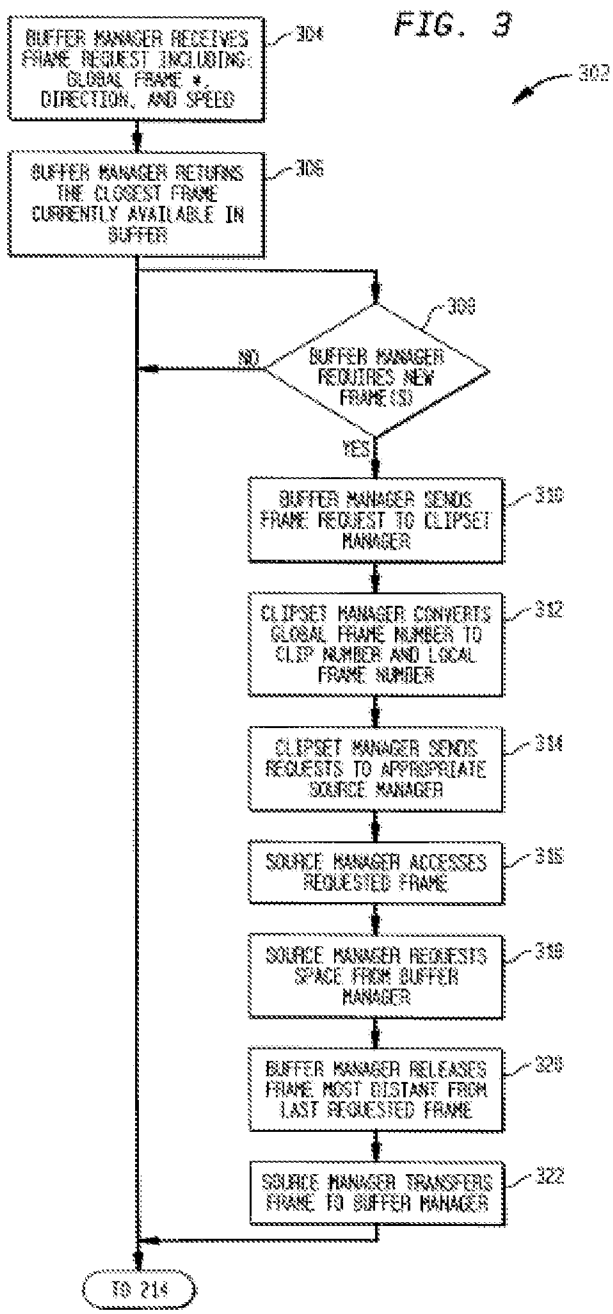


Figure 3 illustrates flowchart 302 for the input/output operation of buffer manager 18.

In reference to Figure 3, Hill illustrates flowchart 302, in which "buffer manager 118 responds to requests for frames from cliplist manager 116 then determines whether new frames need to be loaded" and "[i]f new frames are required, it uses the cliplist manager 116 to retrieve these frames from the appropriate source 122." (Col. 5, ll. 24–28.) In particular, Hill explains that: (i) "[i]n step 304, the buffer manager 118 receives a frame request from cliplist manager 116" such that "[t]he request includes a global frame number" (col. 5, ll. 31–33); (ii) "[i]n step 310, the buffer manager 118 sends requests for new frames to the cliplist manager 116" and "maintains the mapping between global and local frame" (col. 6, ll. 7–10); and (iii) "in step 312, the cliplist manager 116 converts the global frame number to a clip number and local frame number" (col. 6, ll. 10–12).

Because Hill explains that: (i) buffer manager 118 sends requests for new frames to the cliplist manager 116 to retrieve frames from the appropriate source 122; and (ii) the frame request from cliplist manager 116 includes a global frame number, Hill describes the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number."

Patent Owner argues the following:

> [W]hile describing the internal operation of the "source manager" 120 relative to "buffer manager[""] "118" at considerable length, Hill does not disclose how source manager 120 (within the player) actually obtains media data from source 122 (outside the player). In a "black box" manner, relative to the actual media source 122, the disclosure states: "The source manager 120 provides the necessary interface (i.e., hardware or software) for communicating with a source 122." "The source manager 120 in step 316 retrieves the requested frame data

> from the source 122." But there is no disclosure of how media data elements are requested from source 122.

(Appeal Br. 34–35 (citations omitted); *see also* Reply Br. 22–23.) Contrary to Patent Owner's arguments that Hall operates in "a 'black box' manner" with "no disclosure of how media data elements are requested from source 122," as discussed previously, Hall explains that: (i) "[e]ach source 122 . . . may be accessed either locally or via a network server" or "a source may be a site on the Internet" (col. 3, ll. 53–56); and (ii) "the buffer manager 118 receives a frame request from cliplist manager 116," such that "[t]he request includes a global frame number" (col. 5, ll. 31–33).

Therefore, we agree with the Examiner that Hill describes the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number."

We are further unpersuaded by Patent Owner's arguments (Appeal Br. 35; *see also* Reply Br. 23–24) that Hill does not describe the limitation "maintain a record of the serial number of the last media data element that has been received and stored in the player buffer," as recited in independent claim 1.

The Examiner found that buffer manager 118 of Hill, which uses the last global frame number requested for the next request, corresponds to the limitation "maintain a record of the serial number of the last media data element that has been received." (Final Act. 57; *see also* Ans. 27.) We agree with the Examiner's findings.

As discussed previously, in reference to Figure 3, Hill explains that "[i]n step 304, the buffer manager 118 receives a frame request from cliplist manager 116" and "[t]he request includes a global frame number." (Col. 5,

ll. 31–33.) Moreover, Hill explains that "buffer manager 118 normally uses the global frame number last requested as the starting point for determining which frame to request next." (Col. 10, ll. 4–6.) Because Hill explains that: (i) buffer manager 118 receives a frame request from cliplist manager 116; and (ii) buffer manager 118 uses the global frame number last requested as the starting point for determining which frame to request next, Hill describes the limitation "maintain a record of the serial number of the last media data element that has been received and stored in the player buffer."

Patent Owner argues the following:

> Hill discloses individual frame requests in the particular circumstance where it needs catch up the buffer if it is falling behind in streaming. However, it refers to these particular frames as simply one or two "new" frames. To determine which frame to request in that limited circumstance, Hill discloses "adding one to the last request." There is no teaching in Hill to maintain a record of the serial number of the last received segment.

(Appeal Br. 35 (citation and emphases omitted); *see also* Reply Br. 23–24.) Contrary to Patent Owner's arguments, Hill explains that "buffer manager 118 . . . uses the global frame number last requested as the starting point for determining which frame to request next" (col. 10, ll. 4–6) and, as such, Hill describes the limitation "maintain a record of the serial number of the last media data element that has been received and stored in the player buffer."

Therefore, we agree with the Examiner that Hill describes the limitation "maintain a record of the serial number of the last media data element that has been received and stored in the player buffer."

Thus, we sustain the rejection of independent claim 1 under 35 U.S.C. § 102(e).

Dependent Claim 4

Last, we are unpersuaded by Patent Owner's arguments (Appeal Br. 26, 36; *see also* Reply Br. 24–25) that Hill does not describe the limitation "wherein the instructions . . . further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player," as recited in dependent claim 4.

The Examiner found that Figure 4A of Hill, which includes step 480 for determining buffer fill level, corresponds to the limitation "wherein the instructions . . . further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player." (Final Act. 17–18; *see also* Ans. 28–29.) We agree with the Examiner's findings.

Figure 4A of Hill, reproduced below, illustrates the basic operation of a preferred buffer management strategy. (Col. 2, ll. 66–67.)

FIG. 4A

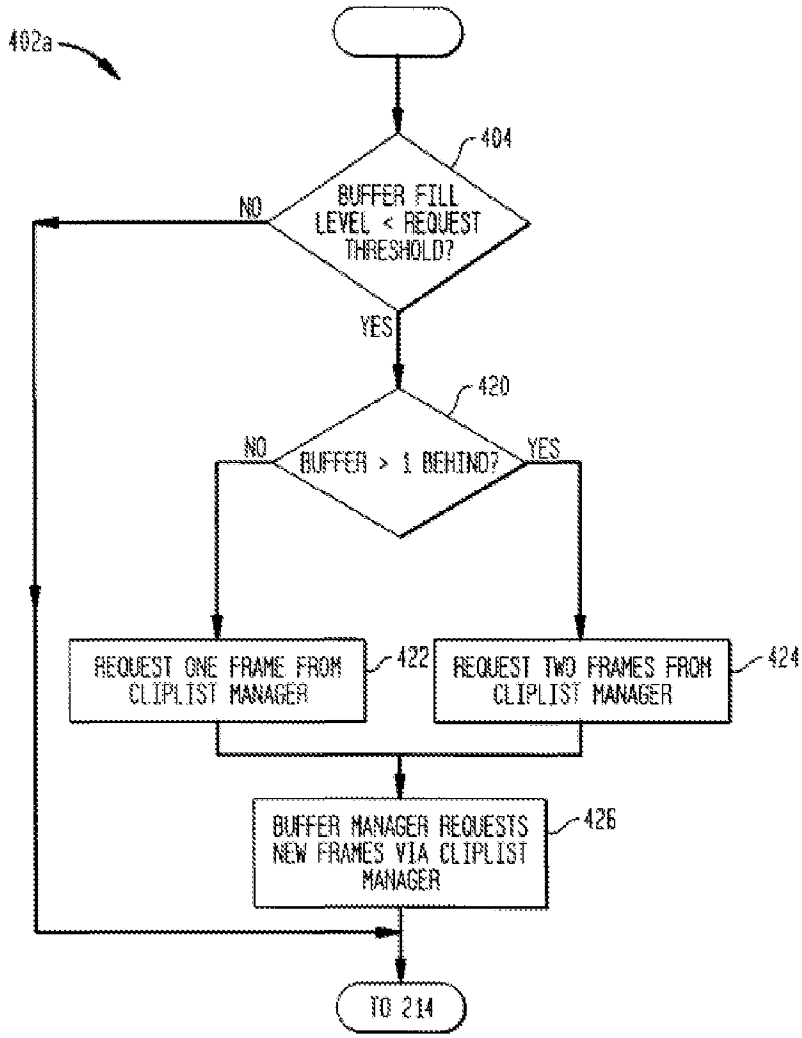


Figure 4A illustrates a flow chart of buffer management strategy.

In reference to Figure 4A, Hill explains that: (i) "the buffer manager 118 determines whether the buffer fill level has dropped below the request threshold, as shown in step 404" (col. 6, ll. 58–60); and (ii) "[i]n step 420, the buffer manager 118 determines by how many frames the request threshold exceeds the fill level" (col. 7, ll. 11–12). With respect to step 420, Hill explains that for a preferred embodiment, "if the play rate is 24 frames

per second, the sources will not be asked to supply more than 48 frames per second." (Col. 7, ll. 17–22.) Because Hill provides for one embodiment in which source 122 supplies 48 frames per second when the play rate is 24 frames per second, Hill describes the limitation "wherein the instructions . . . further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player."

Patent Owner argues the following:

> It is therefore apparent that in the case the buffer is only one frame behind, the request will be for one frame encoded at 24 frames per second. If the request is for two frames, the server will have to supply 48 frames per second—only the latter necessarily being faster than the playback rate. However, there is no teaching that the frames requested one frame at a time should be sent or received any faster than the 24-frame-per-second playback rate. This does not meet the limitation wherein each requested data element is received more rapidly than the playback rate.

(Appeal Br. 36; *see also* Reply Br. 24–25.) Similarly, Patent Owner agues the following:

> The Federal Circuit [in *WAG Acquisition, LLC v. Webpower, Inc.*, 781 F. App'x 1007 (Fed. Cir. 2019)] . . . , addressing a similar limitation in claim 10 of the related 8,122,141 patent, ruled that "the "rate" in claim 10 refers to the rate at which each requested media data element is transmitted from the server to the user computer" and that "[t]he rate limitation in claim 10 therefore refers to the rate at which requested media data elements are sent, not the overall rate at which data is transmitted from the server to the user computer." Thus, the limitation in question was held to be directed to the requested elements, and specifically to "each" requested element, as reflected in the Federal Circuit decision. Claim 4 recites that player instructions cause "the predetermined

> number of media data elements," meaning each of them, to be received at a rate more rapid than the playback rate.

(Appeal Br. 26 (citations omitted).) However, we are not persuaded by Patent Owner's arguments for the same reasons are articulated by another panel of this Board in *WebPower v. WAG Acquisition L.L.C.*, IPR2016-01238 (PTAB July 16, 2020) (Final Written Decision), which addressed the Federal Circuit remand decision, as follows:

> That conclusion [by Patent Owner] relies on an improper importation of an additional limitation into the claim, namely that *all* requested media data elements must be sent by the server at a rate more rapid than the rate at which the streaming media is played back by a user.

(Slip Op. 19 (emphasis in original).)

> That is, Patent Owner appears to read the word "each" in the Federal Circuit's construction as requiring that all media data elements be transmitted faster than the playback rate. As we summarize above, the context in which the federal Circuit arrived at its construction distinguished from our prior construction of "rate" as corresponding to the "overall rate" of transmission from the server to the user computer, such as might be achieved with multiple links over which data elements are sent to the user system. . . . . We discern nothing in the Federal Circuit's decision that compels Patent Owner's implicit additional requirement that *all* media data elements be transmitted faster than the playback rate.

(*Id.* at 20 (emphasis in original).)

Therefore, we agree with the Examiner that Hill describes the limitation "wherein the instructions . . . further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player."

Thus, we sustain the rejection of dependent claim 4 under 35 U.S.C. § 102(e).

Other Pending Rejections

We do not reach the additional rejections of claims 1 and 4 under either: (i) 35 U.S.C. § 102(e) as being anticipated by Shteyn, Carmel, or Imai; and (ii) 35 U.S.C. § 103(a) as being unpatentable over various combinations of Shteyn, Carmel, Glaser, Hill, and Imai. Affirmance of the anticipation-based rejections discussed previously renders it unnecessary to reach the remaining anticipated and obviousness rejections, as all of pending claims have been addressed and found unpatentable. *Cf. In re Gleave*, 560 F.3d 1331, 1338 (Fed. Cir. 2009).

CONCLUSION

The Examiner's decision rejecting claims 1 and 4 under 35 U.S.C. § 102(e) is affirmed.

DECISION SUMMARY

In summary:

Claims Rejected	**35 U.S.C. §**	**Reference(s)/Basis**	**Affirmed**	**Reversed**
1, 4	102(e)[2]	Shteyn		
4	103(a)[3]	Shteyn, Carmel		
4	103(a)[4]	Shteyn, Glaser		
1, 4	102(e)	Hill	1, 4	

[2] Cumulative rejection.
[3] Cumulative rejection.
[4] Cumulative rejection.

4	103(a)[5]	Hill, Carmel		
4	103(e)[6]	Hill, Glaser		
1, 4	103(a)[7]	Shteyn, Hill		
1, 4	102(e)[8]	Carmel		
1, 4	103(a)[9]	Carmel, Hill		
1	102(e)[10]	Imai		
4	103(a)[11]	Imai, Carmel		
4	103(a)[12]	Imai, Glaser		
Overall Outcome			1, 4	

TIME PERIOD FOR RESPONSE

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

[5] Cumulative rejection.
[6] Cumulative rejection.
[7] Cumulative rejection.
[8] Cumulative rejection.
[9] Cumulative rejection.
[10] Cumulative rejection.
[11] Cumulative rejection.
[12] Cumulative rejection.

Appeal 2023-002525
Reexamination Control 90/014,833
Patent 8,327,011 B1

PATENT OWNER/APPELLANT:

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES
231 SOMERVILLE ROAD
BEDMINSTER, NJ 07921

THIRD PARTY REQUESTER:

VANABLE LLP
1290 AVENUE OF THE AMERICAS
NEW YORK, NY 10104-3800

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/014,833	08/25/2021	8327011	125737.538655	1250

7590 04/11/2024

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES, LLC
231 SOMERVILLE ROAD
BEDMINSTER, NJ 07921

EXAMINER	
WASSUM, LUKE S	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
04/11/2024	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte WAG ACQUISITION, LLC
Patent Owner and Appellant

Appeal 2023-002525
Reexamination Control 90/014,833
Patent 8,327,011 B2
Technology Center 3900

Before JOHN A. JEFFERY, ERIC B. CHEN, and BRIAN J. McNAMARA, *Administrative Patent Judges*.

CHEN, *Administrative Patent Judge*.

DECISION ON REQUEST FOR REHEARING

Patent Owner requests rehearing under 37 C.F.R. § 41.52 of our Decision on Appeal entered November 17, 2023 ("Decision" or "Dec."), in which we affirmed the Examiner's final rejection of claims 1 and 4.

The Request for Rehearing is *denied.*

OPINION

First, Patent Owner argues that "the Board made incorrect factual findings concerning Hill and, on that basis, improperly found . . . that Hill discloses . . . 'transmit to the media source a request to send one or more media data elements.'" (Req. Reh'g 1.) In particular, Patent Owner argues the following:

> Hill discloses supplying frames to workstation 104, which frames originate from external sources 122. As disclosed [in Figure 1], this takes place over two distinct path segments, consisting of internal and external portions.

(*Id.* at 5–6.)

> The [Decision] at 7, applying Hill as a matter of anticipation under § 102, has misidentified the relevant "requests," ascribing these to Hill's buffer manager 118, whereas it is clear from Hill at 4:27–34 that it is cliplist manager 116 that makes the requests, and that it does so "via the appropriate source manager 120," which is a component *internal* to workstation 104.

(*Id.* at 6.)

> As noted above, Hill goes no further than to state that "the source manager 120 in step 316 retrieves the requested frame data from source 122." Hill does not say what that request constitutes nor when or how the external retrieval (the one between source manager 120 and the corresponding source 122) is carried out.

(*Id.* at 7.)

Page 7 of our Decision is reproduced below:

> Hill further explains [in reference to Figure 1] that . . . *"[e]ach source 122 may be* any type of digital memory, and may be accessed either locally or via *a network server" or "a source may be a site on the Internet."* (Col. 3, ll. 49–56.) Additionally, Hill also explains that "[t]he buffer manager 118 provides frames upon request to the control unit 114 via the cliplist manager 116" (col. 4, ll. 7–8) and "*[t]he cliplist manager 116 requests frames from a particular source 122 via the appropriate source manager 120*" (col. 4, ll. 27–28).

(Dec. 7 (emphases added).)

Thus, as addressed in our Decision, the limitation "transmit to the media source a request to send one or more media data elements" is broad enough to encompass cliplist manager 116 of Hill, which requests frames from particular source 122 (e.g., network server or Internet site) via the appropriate source manager 120. Patent Owner's arguments with respect to internal and external path segments for the request made by cliplist manager 116—i.e., indirect communication—are not commensurate in scope with claim 1, because the claim does not require a *direct transmission* to the media source.

Second, Patent Owner argues that "the Board made incorrect factual findings concerning Hill and, on that basis, improperly found . . . that Hill discloses . . . [']one or more media data elements, each identified by a serial number.'" (Req. Reh'g 1.) In particular, Patent Owner argues that:

> The [Decision] at 10 (first full paragraph) improperly relies on internal operations of buffer manager 118 and thus improperly conflates Hill's global frame numbers with a local clip frame numbers to find anticipation, notwithstanding the lack of disclosure in Hill that a global frame number can even serve as an identifier of data residing on a particular source 122 (indeed,

> implying that global frame numbers would not be suitable for such identification).
>
> In fact, global frame numbers are not meaningful to external sources 122, which only house clips whose content will be combined into the global presentation. *E.g.*, Hill, Fig. 7 (local frame numbers each starting at 1). At page 11, the [Decision] again looks to a request including a "global frame number," which is not an identification meaningful to external sources 122, and thus cannot correspond to the claimed "requests."

(*Id.* at 7.)

Page 9 of our Decision is reproduced below:

> In reference to Figure 3, Hill illustrates flowchart 302, in which "buffer manager 118 [from Figure 1] responds to requests for frames from cliplist manager 116 then determines whether new frames need to be loaded" and "[i]f new frames are required, it uses the *cliplist manager 116 to retrieve these frames from the appropriate source 122*." (Col. 5, ll. 24–28.) In particular, Hill explains that: (i) "[i]n step 304, *the buffer manager 118 receives a frame request from cliplist manager 116*" such that "*[t]he request includes a global frame number*" (col. 5, ll. 31–33); (ii) "[i]n step 310, the buffer manager 118 sends requests for new frames to the cliplist manager 116" and "maintains the mapping between global and local frame" (col. 6, ll. 7–10); and (iii) "in step 312, *the cliplist manager 116 converts the global frame number to a clip number and local frame number*" (col. 6, ll. 10–12).

(Dec. 9 (emphases added).) Thus, as addressed in our Decision, the limitation "one or more media data elements, each identified by a serial number" is broad enough to encompass the functionality of cliplist manager 116 of Hill, which requests frames, such that the request also includes a global frame number. Contrary to Patent Owner's arguments that "global frame numbers are not meaningful to external sources 122" and "a 'global frame number,' . . . is not an identification meaningful to external

sources 122" (Req. Reh'g 7), Hill expressly states that "cliplist manager 116 . . . retrieve[s] these frames from the appropriate source 122" (col. 5, ll. 27–28) and "[t]he request [from cliplist manager 116] includes a global frame number" (col. 5, ll. 31–33).

Third, Patent Owner argues the following:

> To the extent that inherency in such external communications is relied upon, the Board cites no factual underpinning as to why any undisclosed request in Hill must necessarily include a requested frame number. It is uncontested that Hill is silent on the issue as to how requests for media are made to sources 122, disclosing only that "source manager 120 in step 316 retrieves the requested frame data from the source 122" (Hill, 6:14–16), and nothing more. Given that inherency is a question of fact, without any factual support, any inherency argument must fail.

(Req. Reh'g 8.)

Page 9 of our Decision is reproduced below:

> Because Hill explains that: (i) buffer manager 118 sends requests for new frames to the cliplist manager 116 to retrieve frames from the appropriate source 122; and (ii) the frame request from cliplist manager 116 includes a global frame number, Hill describes the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number."

(Dec. 9.) Thus, our Decision did not rely upon inherency, because as discussed previously, Hill *expressly* discloses the limitation "the limitation "transmit to the media source a request to send one or more media data elements."

Last, with respect to the limitation "maintain a record of the serial number of the last media data element that has been received," Patent Owner argues the following:

> The [Decision] relies on disclosure in Hill that "buffer manager 118 uses the global frame number last requested as the starting point for determining which frame to request next." But, as noted above, buffer manager 118 does not communicate with external source 122 at all. . . . There is no determinable relationship between that and knowing the serial number (clip number and local frame number) of the last frame that has been received "by the media player."

(Req. Reh'g 9 (citation omitted).) We are not persuaded by these arguments for the reasons discussed previously with respect to the limitation "transmit to the media source a request to send one or more media data elements."

CONCLUSION

The Request for Rehearing has been considered and is *denied*.

Outcome of Decision on Rehearing:

Claim(s) Rejected	**35 U.S.C. §**	**Reference(s)/Basis**	**Denied**	**Granted**
1, 4	102	Hill	1, 4	

Final Outcome of Appeal after Rehearing:

Claim(s) Rejected	**35 U.S.C. §**	**Reference(s)/Basis**	**Affirmed**	**Reversed**
1, 4	102	Hill	1, 4	

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv). *See* 37 C.F.R. § 41.50(f).

DENIED

Appeal 2023-002525
Reexamination Control 90/014,833
Patent 8,327,011 B2

PATENT OWNER/APPELLANT:

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES
231 SOMERVILLEROAD
BEDMINSTER, NJ 07921

THIRD PARTY REQUESTER:

VANABLE LLP
1290 AVENUE OF THE AMERICAS
NEW YORK, NY 10104-3800

UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

In re WAG Acquisition, LLC	)	
	)	
Serial No. 90/014,833	)	**Appeal No: 24-1971**
	)	

NOTICE FORWARDING CERTIFIED LIST

A notice of appeal to the United States Court of Appeals for the Federal Circuit was timely filed on June 13, 2024, in the Patent and Trademark Office in connection with the above-identified **ex parte re-examination** proceeding. Pursuant to 35 U.S.C. § 143 and Federal Circuit Rule 17(b)(1), a certified list is this day being forwarded to the Federal Circuit.

Brian Racilla is the attorney representing the Director in this appeal. Counsel for appellant must contact the Solicitor's Office at 571-272-9035 to arrange for designating the record.

Respectfully submitted,

Under Secretary of Commerce for
Intellectual Property and Director of the
United States Patent and Trademark Office

Date: July 30, 2024

By: */s/Natasha M. Brotten*
Natasha M. Brotten
Paralegal Specialist
Mail Stop 8
P.O. Box 1450
Alexandria, VA 22313-1450
571-272-9035

CERTIFICATE OF SERVICE

The undersigned hereby certifies that a true and correct copy of the foregoing **NOTICE FORWARDING CERTIFIED LIST** has been served, via electronic mail, on counsel for appellant this 30th day of July, 2024 as follows:

Ronald Abramson
ron.abramson@listonabramson.com

Ari J. Jaffess
ari.jaffess@listonabramson.com

By: */s/Natasha M. Brotten*

Natasha M. Brotten
Paralegal Specialist

Form PTO 55 (12-80)

U.S. DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

July 30, 2024
(Date)

THIS IS TO CERTIFY that the attached document is a list of the contents comprised from the electronic file of the Patent Re-examination Application identified below; said contents being a list of the documents comprising the record before the United States Patent and Trademark Office for the Reexamination Proceeding of:

Patent Owner(s): **WAG Acquisition, LLC; Streamray Inc. c/o Frank M. Gaspro; Friendfinder Networks Inc.**

Reexamination No.: **90/014,833**

Filed: **08/25/2021**

Title of Invention: **Streaming Media Buffering System**

By authority of the

DIRECTOR OF THE UNITED STATES
PATENT AND TRADEMARK OFFICE

/s/*Natasha M. Brotten*

Certifying Officer

Prosecution History Ser. No. 90/014,833

Date	Document
08/25/2021	Request for Ex Parte Reexamination of U.S. Patent No. 8,327,011
08/25/2021	Information Disclosure Statement
08/25/2021	Fee Worksheet
08/30/2021	Notice of Assignment of Reexamination Request
08/30/2021	Notice of Reexamination Request Filing Date
08/30/2021	Bibliographic Data Sheet
09/20/2021	Examiner Interview Summary Record
11/16/2021	Order Granting Request for Ex Parte Reexamination
03/04/2022	Non-Final Office Action
04/22/2022	Request for Extension of Time
04/22/2022	Fee Worksheet
04/25/2022	Order Granting Extension of Time Period for Response
04/25/2022	Submission of Fees for Extension of Time
04/25/2022	Refund Request
04/25/2022	Fee Worksheet
04/25/2022	Fee Worksheet
04/25/2022	Fee Worksheet
06/06/2022	Response After Non-Final Office Action
07/27/2022	Final Office Action
09/28/2022	Petition for Revival of an Application for Patent Abandoned Unintentionally
09/28/2022	Fee Worksheet
09/28/2022	Notice of Appeal to the Patent Trial and Appeal Board
10/03/2022	Petition Decision
11/28/2022	Appeal Brief
03/15/2023	Examiner's Answer
05/15/2023	Reply Brief
05/15/2023	Request for Oral Hearing
05/15/2023	Fee Worksheet
05/23/2023	Patent Trial and Appeal Board Docketing Notice
07/31/2023	Notification of Appeal Hearing
09/12/2023	Confirmation of Hearing by Appellant
09/13/2023	Decision Granting Video Hearing
10/05/2023	Patent Trial and Appeal Board (PTAB) Oral Hearing Transcript
11/17/2023	Decision on Appeal
01/16/2024	Request for Rehearing
02/16/2024	Mail returned to USPTO as undelivered
04/11/2024	Decision on Request for Rehearing
06/13/2024	Appeal to Court of Appeals for the Federal Circuit

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/014,833	08/25/2021	8327011	125737.538655	1250

7590 11/17/2023

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES, LLC
231 SOMERVILLE ROAD
BEDMINSTER, NJ 07921

EXAMINER	
WASSUM, LUKE S	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
11/17/2023	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte WAG ACQUISITION, LLC
Patent Owner and Appellant

Appeal 2023-002525
Reexamination Control 90/014,833
Patent 8,327,011 B2
Technology Center 3900

Before JOHN A. JEFFERY, ERIC B. CHEN, and BRIAN J. McNAMARA, *Administrative Patent Judges*.

CHEN, *Administrative Patent Judge*.

DECISION ON APPEAL

Pursuant to 35 U.S.C. §§ 134(b) and 306, Patent Owner[1] appeals from the final rejection of claims 1 and 4. Claims 2 and 3 are not subject to reexamination.

A video oral hearing was held on September 18, 2023. The record includes a written transcript of the oral hearing. We have jurisdiction under 35 U.S.C. § 6(b).

We AFFIRM.

---

[1] Patent Owner identifies the real party in interest as WAG Acquisition, LLC. (Appeal Br. 1.)

STATEMENT OF THE CASE

Reexamination Proceedings

A request for *ex parte* reexamination of U.S. Patent No. 8,327,011 B2 ("the '011 patent") was filed on August 25, 2021 and assigned Control No. 90/014,833. The '011 patent, entitled "Streaming Media Buffering System" issued December 4, 2012 to Harold Edward Price, based on Application No. 13/374,942, filed January 24, 2012. The '011 patent is part of a series of multiple continuing applications, with the first application filed on March 28, 2001, which claims priority to provisional Application No. 60/231,997, filed on September 12, 2000.

Claimed Subject Matter

The claims are directed to streaming media, sent via the Internet, such that the media in the user buffer accumulates and immediately played on a user's computer. (Abstract.)

Related Litigation

The '011 patent has been asserted in the following litigation: (i) *WAG Acquisition, LLC v. Sobonito Investments, Ltd. et al.*, No. 2:14-cv-1661 (D.N.J.) (dismissed); (ii) *WAG Acquisition, LLC v. Multi Media, LLC et al.*, No. 2:14-cv-2340 (D.N.J.) (dismissed); (iii) *WAG Acquisition, LLC v. Data Conversions, Inc. et al.*, No. 2:14-cv-2345 (D.N.J.) (dismissed); (iv) *WAG Acquisition, LLC v. Flying Crocodile, Inc. et al.*, No. 2:14-cv-2674 (D.N.J.); (v) *WAG Acquisition, LLC v. Gattyán Group S.á.r.l. et al.*, No. 2:14-cv-2832 (D.N.J.) (dismissed); (vi) *WAG Acquisition, LLC v. FriendFinder Networks Inc., et al.*, No. 2:14-cv-3456 (D.N.J.); (vii) *WAG Acquisition,*

LLC v. Vubeology, Inc. et al., No. 2:14-cv-4531 (D.N.J.) (dismissed); (viii) *WAG Acquisition, LLC v. Gamelink Int'l Ltd. et al.*, No. 2:15-cv-3416 (D.N.J.) (dismissed); and (ix) *WAG Acquisition LLC v. WebPower, Inc. et al.*, No. 2:15-cv-3581 (D.N.J.) (dismissed).

The '011 patent was subject to the following petitions for *inter partes* reviews: (i) *FriendFinder Networks Inc. et al. v. WAG Acquisition LLC*, IPR2015-01033 (PTAB Oct. 19, 2015) (institution denied); (ii) *I.M.L. SLU v. WAG Acquisition LLC*, IPR2016-01655 (PTAB Feb. 27, 2017) (institution denied); and (iii) *WebPower v. WAG Acquisition LLC*, IPR2016-01161 (PTAB Dec. 12, 2016) (institution denied).

The Claims

Claims 1 and 4 are illustrative of the claimed subject matter, and reproduced below with disputed limitations in italics:

> 1. A media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number, comprising
>
> a processor;
>
> a memory;
>
> a connection to the network; and
>
> media player software comprising
>
> instructions to cause the media player to request from the media source a predetermined number of media data elements;
>
> instructions to cause the media player to receive media data elements sent to the media player by the

media source and store the media data elements in the memory;

instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to *maintain a record of the serial number of the last media data element that has been received and stored in the player buffer*;

instructions to cause the media player to play media data elements sequentially from the player buffer; and

instructions to cause the media player to *transmit to the media source a request to send one or more media data elements, each identified by a serial number*, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.

4. The media player of claim 1, *wherein the instructions* for causing the media player to request from the media source a predetermined number of media data elements *further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player*.

REFERENCES

Name	Reference	Date
Glaser et al.	US 5,793,980	Aug. 11, 1998
Imai et al.	US 5,987,510	Nov. 16, 1999
Hill	US 6,005,600	Dec. 21, 1999
Carmel et al.	US 6,389,473 B1	May 14, 2002
Shteyn	US 7,529,806 B1	May 5, 2009

REJECTIONS

A. Claims 1 and 4 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Shteyn.

B. Claim 4 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Shteyn and Carmel.

C. Claim 4 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Shteyn and Glaser.

D. Claims 1 and 4 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Hill.

E. Claim 4 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Hill and Carmel.

F. Claim 4 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Hill and Glaser.

G. Claims 1 and 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Shteyn and Hill.

H. Claims 1 and 4 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Carmel.

I. Claims 1 and 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Carmel and Hill.

J. Claim 1 stands rejected under 35 U.S.C. § 102(e) as being anticipated by Imai.

K. Claims 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Imai and Carmel.

L. Claims 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Imai and Glaser.

OPINION

§ 102 Rejection—Hill

Independent Claim 1

We are unpersuaded by Patent Owner's arguments (Appeal Br. 34–35; *see also* Reply Br. 22–23) that Hill does not describe the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number," as recited in independent claim 1.

The Examiner found that buffer manager 118 of Hill, which retrieves files from source 122, corresponds to the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number." (Final Act. 15–16; *see also* Ans. 25–26.) We agree with the Examiner's findings.

Hill relates "to playing movie data held in various storage devices with varying bandwidth capabilities, where the movie player adapts to these varying conditions." (Col. 1, ll. 7–10.) Figure 1 of Hill, reproduced below, illustrates digital network environment 102. (Col. 3, ll. 27–28.)

FIG. 1

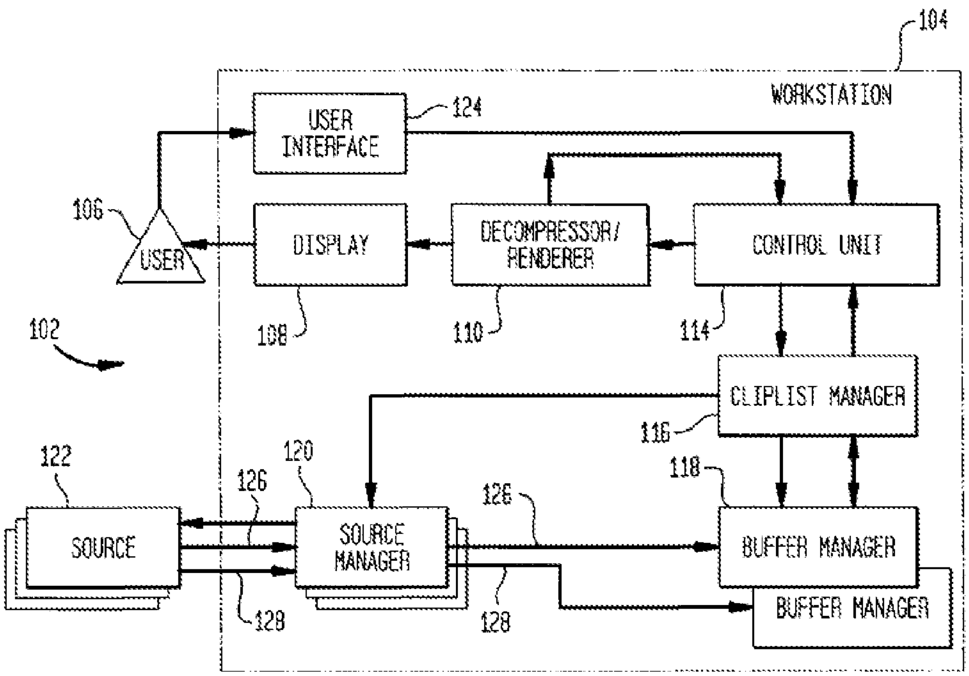


Figure 1 illustrates digital network environment 102.

In reference to Figure 1, Hill explains that digital network environment 102 features "[a] user 106 [that] utilizes the workstation 104 to play movie data." (Col. 3, ll. 27–30.) Hill further explains that "the movie data of interest to the user contains N_c clips, each stored in a separate source 122," such that "[e]ach source 122 may be any type of digital memory, and may be accessed either locally or via a network server" or "a source may be a site on the Internet." (Col. 3, ll. 49–56.) Additionally, Hill also explains that "[t]he buffer manager 118 provides frames upon request to the control unit 114 via the cliplist manager 116" (col. 4, ll. 7–8) and "[t]he cliplist manager 116 requests frames from a particular source 122 via the appropriate source manager 120" (col. 4, ll. 27–28).

Appeal 2023-002525
Reexamination Control 90/014,833
Patent 8,327,011 B1

Figure 3 of Hill, reproduced below, illustrates flowchart 320 of the input/output operation for buffer manager 18. (Col. 2, ll. 64–65.)

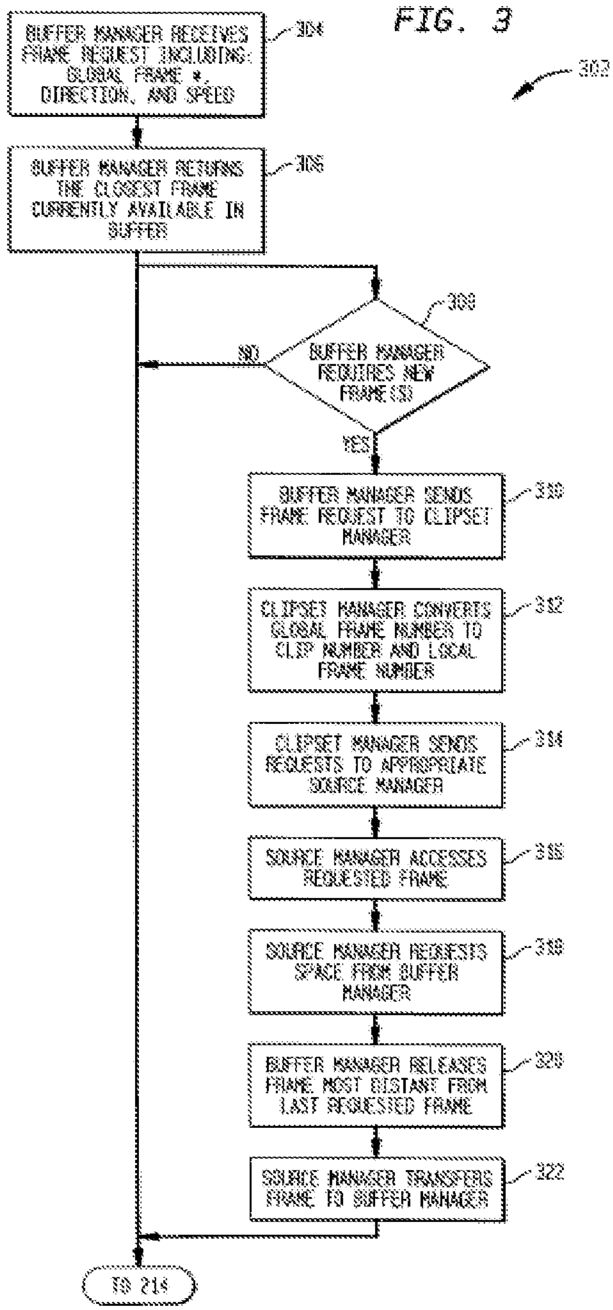


Figure 3 illustrates flowchart 302 for the input/output operation of buffer manager 18.

In reference to Figure 3, Hill illustrates flowchart 302, in which "buffer manager 118 responds to requests for frames from cliplist manager 116 then determines whether new frames need to be loaded" and "[i]f new frames are required, it uses the cliplist manager 116 to retrieve these frames from the appropriate source 122." (Col. 5, ll. 24–28.) In particular, Hill explains that: (i) "[i]n step 304, the buffer manager 118 receives a frame request from cliplist manager 116" such that "[t]he request includes a global frame number" (col. 5, ll. 31–33); (ii) "[i]n step 310, the buffer manager 118 sends requests for new frames to the cliplist manager 116" and "maintains the mapping between global and local frame" (col. 6, ll. 7–10); and (iii) "in step 312, the cliplist manager 116 converts the global frame number to a clip number and local frame number" (col. 6, ll. 10–12).

Because Hill explains that: (i) buffer manager 118 sends requests for new frames to the cliplist manager 116 to retrieve frames from the appropriate source 122; and (ii) the frame request from cliplist manager 116 includes a global frame number, Hill describes the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number."

Patent Owner argues the following:

> [W]hile describing the internal operation of the "source manager" 120 relative to "buffer manager["] "118" at considerable length, Hill does not disclose how source manager 120 (within the player) actually obtains media data from source 122 (outside the player). In a "black box" manner, relative to the actual media source 122, the disclosure states: "The source manager 120 provides the necessary interface (i.e., hardware or software) for communicating with a source 122." "The source manager 120 in step 316 retrieves the requested frame data

> from the source 122." But there is no disclosure of how media data elements are requested from source 122.

(Appeal Br. 34–35 (citations omitted); *see also* Reply Br. 22–23.) Contrary to Patent Owner's arguments that Hall operates in "a 'black box' manner" with "no disclosure of how media data elements are requested from source 122," as discussed previously, Hall explains that: (i) "[e]ach source 122 . . . may be accessed either locally or via a network server" or "a source may be a site on the Internet" (col. 3, ll. 53–56); and (ii) "the buffer manager 118 receives a frame request from cliplist manager 116," such that "[t]he request includes a global frame number" (col. 5, ll. 31–33).

Therefore, we agree with the Examiner that Hill describes the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number."

We are further unpersuaded by Patent Owner's arguments (Appeal Br. 35; *see also* Reply Br. 23–24) that Hill does not describe the limitation "maintain a record of the serial number of the last media data element that has been received and stored in the player buffer," as recited in independent claim 1.

The Examiner found that buffer manager 118 of Hill, which uses the last global frame number requested for the next request, corresponds to the limitation "maintain a record of the serial number of the last media data element that has been received." (Final Act. 57; *see also* Ans. 27.) We agree with the Examiner's findings.

As discussed previously, in reference to Figure 3, Hill explains that "[i]n step 304, the buffer manager 118 receives a frame request from cliplist manager 116" and "[t]he request includes a global frame number." (Col. 5,

ll. 31–33.) Moreover, Hill explains that "buffer manager 118 normally uses the global frame number last requested as the starting point for determining which frame to request next." (Col. 10, ll. 4–6.) Because Hill explains that: (i) buffer manager 118 receives a frame request from cliplist manager 116; and (ii) buffer manager 118 uses the global frame number last requested as the starting point for determining which frame to request next, Hill describes the limitation "maintain a record of the serial number of the last media data element that has been received and stored in the player buffer."

Patent Owner argues the following:

> Hill discloses individual frame requests in the particular circumstance where it needs catch up the buffer if it is falling behind in streaming. However, it refers to these particular frames as simply one or two "new" frames. To determine which frame to request in that limited circumstance, Hill discloses "adding one to the last request." There is no teaching in Hill to maintain a record of the serial number of the last received segment.

(Appeal Br. 35 (citation and emphases omitted); *see also* Reply Br. 23–24.) Contrary to Patent Owner's arguments, Hill explains that "buffer manager 118 . . . uses the global frame number last requested as the starting point for determining which frame to request next" (col. 10, ll. 4–6) and, as such, Hill describes the limitation "maintain a record of the serial number of the last media data element that has been received and stored in the player buffer."

Therefore, we agree with the Examiner that Hill describes the limitation "maintain a record of the serial number of the last media data element that has been received and stored in the player buffer."

Thus, we sustain the rejection of independent claim 1 under 35 U.S.C. § 102(e).

Dependent Claim 4

Last, we are unpersuaded by Patent Owner's arguments (Appeal Br. 26, 36; *see also* Reply Br. 24–25) that Hill does not describe the limitation "wherein the instructions . . . further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player," as recited in dependent claim 4.

The Examiner found that Figure 4A of Hill, which includes step 480 for determining buffer fill level, corresponds to the limitation "wherein the instructions . . . further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player." (Final Act. 17–18; *see also* Ans. 28–29.) We agree with the Examiner's findings.

Figure 4A of Hill, reproduced below, illustrates the basic operation of a preferred buffer management strategy. (Col. 2, ll. 66–67.)

FIG. 4A

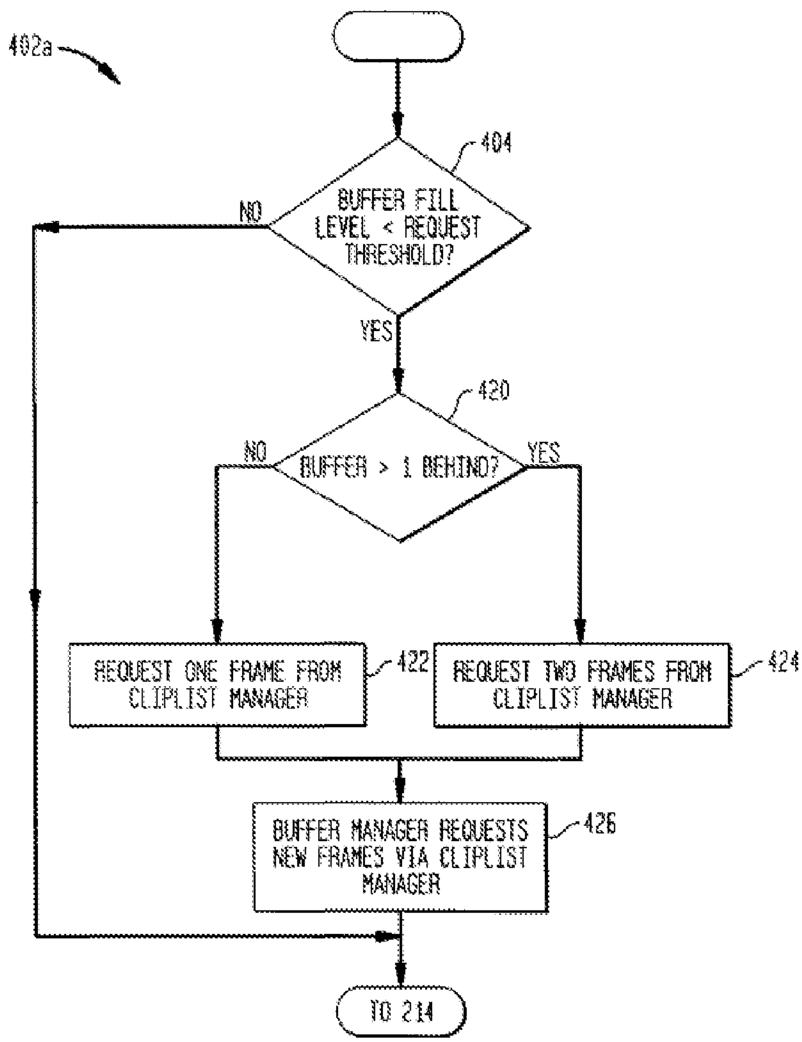


Figure 4A illustrates a flow chart of buffer management strategy.

In reference to Figure 4A, Hill explains that: (i) "the buffer manager 118 determines whether the buffer fill level has dropped below the request threshold, as shown in step 404" (col. 6, ll. 58–60); and (ii) "[i]n step 420, the buffer manager 118 determines by how many frames the request threshold exceeds the fill level" (col. 7, ll. 11–12). With respect to step 420, Hill explains that for a preferred embodiment, "if the play rate is 24 frames

per second, the sources will not be asked to supply more than 48 frames per second." (Col. 7, ll. 17–22.) Because Hill provides for one embodiment in which source 122 supplies 48 frames per second when the play rate is 24 frames per second, Hill describes the limitation "wherein the instructions . . . further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player."

Patent Owner argues the following:

> It is therefore apparent that in the case the buffer is only one frame behind, the request will be for one frame encoded at 24 frames per second. If the request is for two frames, the server will have to supply 48 frames per second—only the latter necessarily being faster than the playback rate. However, there is no teaching that the frames requested one frame at a time should be sent or received any faster than the 24-frame-per-second playback rate. This does not meet the limitation wherein each requested data element is received more rapidly than the playback rate.

(Appeal Br. 36; *see also* Reply Br. 24–25.) Similarly, Patent Owner agues the following:

> The Federal Circuit [in *WAG Acquisition, LLC v. Webpower, Inc.*, 781 F. App'x 1007 (Fed. Cir. 2019)] . . . , addressing a similar limitation in claim 10 of the related 8,122,141 patent, ruled that "the "rate" in claim 10 refers to the rate at which each requested media data element is transmitted from the server to the user computer" and that "[t]he rate limitation in claim 10 therefore refers to the rate at which requested media data elements are sent, not the overall rate at which data is transmitted from the server to the user computer." Thus, the limitation in question was held to be directed to the requested elements, and specifically to "each" requested element, as reflected in the Federal Circuit decision. Claim 4 recites that player instructions cause "the predetermined

> number of media data elements," meaning each of them, to be received at a rate more rapid than the playback rate.

(Appeal Br. 26 (citations omitted).) However, we are not persuaded by Patent Owner's arguments for the same reasons are articulated by another panel of this Board in *WebPower v. WAG Acquisition L.L.C.*, IPR2016-01238 (PTAB July 16, 2020) (Final Written Decision), which addressed the Federal Circuit remand decision, as follows:

> That conclusion [by Patent Owner] relies on an improper importation of an additional limitation into the claim, namely that *all* requested media data elements must be sent by the server at a rate more rapid than the rate at which the streaming media is played back by a user.

(Slip Op. 19 (emphasis in original).)

> That is, Patent Owner appears to read the word "each" in the Federal Circuit's construction as requiring that all media data elements be transmitted faster than the playback rate. As we summarize above, the context in which the federal Circuit arrived at its construction distinguished from our prior construction of "rate" as corresponding to the "overall rate" of transmission from the server to the user computer, such as might be achieved with multiple links over which data elements are sent to the user system. . . . . We discern nothing in the Federal Circuit's decision that compels Patent Owner's implicit additional requirement that *all* media data elements be transmitted faster than the playback rate.

(*Id.* at 20 (emphasis in original).)

Therefore, we agree with the Examiner that Hill describes the limitation "wherein the instructions . . . further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player."

Thus, we sustain the rejection of dependent claim 4 under 35 U.S.C. § 102(e).

Other Pending Rejections

We do not reach the additional rejections of claims 1 and 4 under either: (i) 35 U.S.C. § 102(e) as being anticipated by Shteyn, Carmel, or Imai; and (ii) 35 U.S.C. § 103(a) as being unpatentable over various combinations of Shteyn, Carmel, Glaser, Hill, and Imai. Affirmance of the anticipation-based rejections discussed previously renders it unnecessary to reach the remaining anticipated and obviousness rejections, as all of pending claims have been addressed and found unpatentable. *Cf. In re Gleave*, 560 F.3d 1331, 1338 (Fed. Cir. 2009).

CONCLUSION

The Examiner's decision rejecting claims 1 and 4 under 35 U.S.C. § 102(e) is affirmed.

DECISION SUMMARY

In summary:

Claims Rejected	**35 U.S.C. §**	**Reference(s)/Basis**	**Affirmed**	**Reversed**
1, 4	102(e)[2]	Shteyn		
4	103(a)[3]	Shteyn, Carmel		
4	103(a)[4]	Shteyn, Glaser		
1, 4	102(e)	Hill	1, 4	

[2] Cumulative rejection.
[3] Cumulative rejection.
[4] Cumulative rejection.

4	103(a)[5]	Hill, Carmel		
4	103(e)[6]	Hill, Glaser		
1, 4	103(a)[7]	Shteyn, Hill		
1, 4	102(e)[8]	Carmel		
1, 4	103(a)[9]	Carmel, Hill		
1	102(e)[10]	Imai		
4	103(a)[11]	Imai, Carmel		
4	103(a)[12]	Imai, Glaser		
Overall Outcome			1, 4	

TIME PERIOD FOR RESPONSE

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

[5] Cumulative rejection.
[6] Cumulative rejection.
[7] Cumulative rejection.
[8] Cumulative rejection.
[9] Cumulative rejection.
[10] Cumulative rejection.
[11] Cumulative rejection.
[12] Cumulative rejection.

Appeal 2023-002525
Reexamination Control 90/014,833
Patent 8,327,011 B1

PATENT OWNER/APPELLANT:

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES
231 SOMERVILLE ROAD
BEDMINSTER, NJ 07921

THIRD PARTY REQUESTER:

VANABLE LLP
1290 AVENUE OF THE AMERICAS
NEW YORK, NY 10104-3800

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/014,833	08/25/2021	8327011	125737.538655	1250

7590 04/11/2024

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES, LLC
231 SOMERVILLE ROAD
BEDMINSTER, NJ 07921

EXAMINER	
WASSUM, LUKE S	
ART UNIT	**PAPER NUMBER**
3992	
MAIL DATE	**DELIVERY MODE**
04/11/2024	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte WAG ACQUISITION, LLC
Patent Owner and Appellant

Appeal 2023-002525
Reexamination Control 90/014,833
Patent 8,327,011 B2
Technology Center 3900

Before JOHN A. JEFFERY, ERIC B. CHEN, and BRIAN J. McNAMARA, *Administrative Patent Judges*.

CHEN, *Administrative Patent Judge*.

DECISION ON REQUEST FOR REHEARING

Patent Owner requests rehearing under 37 C.F.R. § 41.52 of our Decision on Appeal entered November 17, 2023 ("Decision" or "Dec."), in which we affirmed the Examiner's final rejection of claims 1 and 4.

The Request for Rehearing is *denied.*

OPINION

First, Patent Owner argues that "the Board made incorrect factual findings concerning Hill and, on that basis, improperly found . . . that Hill discloses . . . 'transmit to the media source a request to send one or more media data elements.'" (Req. Reh'g 1.) In particular, Patent Owner argues the following:

> Hill discloses supplying frames to workstation 104, which frames originate from external sources 122. As disclosed [in Figure 1], this takes place over two distinct path segments, consisting of internal and external portions.

(*Id.* at 5–6.)

> The [Decision] at 7, applying Hill as a matter of anticipation under § 102, has misidentified the relevant "requests," ascribing these to Hill's buffer manager 118, whereas it is clear from Hill at 4:27–34 that it is cliplist manager 116 that makes the requests, and that it does so "via the appropriate source manager 120," which is a component *internal* to workstation 104.

(*Id.* at 6.)

> As noted above, Hill goes no further than to state that "the source manager 120 in step 316 retrieves the requested frame data from source 122." Hill does not say what that request constitutes nor when or how the external retrieval (the one between source manager 120 and the corresponding source 122) is carried out.

(*Id.* at 7.)

Page 7 of our Decision is reproduced below:

> Hill further explains [in reference to Figure 1] that . . . *"[e]ach source 122 may be* any type of digital memory, and may be accessed either locally or via *a network server" or "a source may be a site on the Internet."* (Col. 3, ll. 49–56.) Additionally, Hill also explains that "[t]he buffer manager 118 provides frames upon request to the control unit 114 via the cliplist manager 116" (col. 4, ll. 7–8) and "*[t]he cliplist manager 116 requests frames from a particular source 122 via the appropriate source manager 120*" (col. 4, ll. 27–28).

(Dec. 7 (emphases added).)

Thus, as addressed in our Decision, the limitation "transmit to the media source a request to send one or more media data elements" is broad enough to encompass cliplist manager 116 of Hill, which requests frames from particular source 122 (e.g., network server or Internet site) via the appropriate source manager 120. Patent Owner's arguments with respect to internal and external path segments for the request made by cliplist manager 116—i.e., indirect communication—are not commensurate in scope with claim 1, because the claim does not require a *direct transmission* to the media source.

Second, Patent Owner argues that "the Board made incorrect factual findings concerning Hill and, on that basis, improperly found . . . that Hill discloses . . . [']one or more media data elements, each identified by a serial number.'" (Req. Reh'g 1.) In particular, Patent Owner argues that:

> The [Decision] at 10 (first full paragraph) improperly relies on internal operations of buffer manager 118 and thus improperly conflates Hill's global frame numbers with a local clip frame numbers to find anticipation, notwithstanding the lack of disclosure in Hill that a global frame number can even serve as an identifier of data residing on a particular source 122 (indeed,

> implying that global frame numbers would not be suitable for such identification).
>
> In fact, global frame numbers are not meaningful to external sources 122, which only house clips whose content will be combined into the global presentation. *E.g.*, Hill, Fig. 7 (local frame numbers each starting at 1). At page 11, the [Decision] again looks to a request including a "global frame number," which is not an identification meaningful to external sources 122, and thus cannot correspond to the claimed "requests."

(*Id.* at 7.)

Page 9 of our Decision is reproduced below:

> In reference to Figure 3, Hill illustrates flowchart 302, in which "buffer manager 118 [from Figure 1] responds to requests for frames from cliplist manager 116 then determines whether new frames need to be loaded" and "[i]f new frames are required, it uses the *cliplist manager 116 to retrieve these frames from the appropriate source 122*." (Col. 5, ll. 24–28.) In particular, Hill explains that: (i) "[i]n step 304, *the buffer manager 118 receives a frame request from cliplist manager 116*" such that "*[t]he request includes a global frame number*" (col. 5, ll. 31–33); (ii) "[i]n step 310, the buffer manager 118 sends requests for new frames to the cliplist manager 116" and "maintains the mapping between global and local frame" (col. 6, ll. 7–10); and (iii) "in step 312, *the cliplist manager 116 converts the global frame number to a clip number and local frame number*" (col. 6, ll. 10–12).

(Dec. 9 (emphases added).) Thus, as addressed in our Decision, the limitation "one or more media data elements, each identified by a serial number" is broad enough to encompass the functionality of cliplist manager 116 of Hill, which requests frames, such that the request also includes a global frame number. Contrary to Patent Owner's arguments that "global frame numbers are not meaningful to external sources 122" and "a 'global frame number,' . . . is not an identification meaningful to external

sources 122" (Req. Reh'g 7), Hill expressly states that "cliplist manager 116 . . . retrieve[s] these frames from the appropriate source 122" (col. 5, ll. 27–28) and "[t]he request [from cliplist manager 116] includes a global frame number" (col. 5, ll. 31–33).

Third, Patent Owner argues the following:

> To the extent that inherency in such external communications is relied upon, the Board cites no factual underpinning as to why any undisclosed request in Hill must necessarily include a requested frame number. It is uncontested that Hill is silent on the issue as to how requests for media are made to sources 122, disclosing only that "source manager 120 in step 316 retrieves the requested frame data from the source 122" (Hill, 6:14–16), and nothing more. Given that inherency is a question of fact, without any factual support, any inherency argument must fail.

(Req. Reh'g 8.)

Page 9 of our Decision is reproduced below:

> Because Hill explains that: (i) buffer manager 118 sends requests for new frames to the cliplist manager 116 to retrieve frames from the appropriate source 122; and (ii) the frame request from cliplist manager 116 includes a global frame number, Hill describes the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number."

(Dec. 9.) Thus, our Decision did not rely upon inherency, because as discussed previously, Hill *expressly* discloses the limitation "the limitation "transmit to the media source a request to send one or more media data elements."

Last, with respect to the limitation "maintain a record of the serial number of the last media data element that has been received," Patent Owner argues the following:

> The [Decision] relies on disclosure in Hill that "buffer manager 118 uses the global frame number last requested as the starting point for determining which frame to request next." But, as noted above, buffer manager 118 does not communicate with external source 122 at all. . . . There is no determinable relationship between that and knowing the serial number (clip number and local frame number) of the last frame that has been received "by the media player."

(Req. Reh'g 9 (citation omitted).) We are not persuaded by these arguments for the reasons discussed previously with respect to the limitation "transmit to the media source a request to send one or more media data elements."

CONCLUSION

The Request for Rehearing has been considered and is *denied*.

Outcome of Decision on Rehearing:

Claim(s) Rejected	35 U.S.C. §	Reference(s)/Basis	Denied	Granted
1, 4	102	Hill	1, 4	

Final Outcome of Appeal after Rehearing:

Claim(s) Rejected	35 U.S.C. §	Reference(s)/Basis	Affirmed	Reversed
1, 4	102	Hill	1, 4	

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv). *See* 37 C.F.R. § 41.50(f).

DENIED

Appeal 2023-002525
Reexamination Control 90/014,833
Patent 8,327,011 B2

PATENT OWNER/APPELLANT:

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES
231 SOMERVILLEROAD
BEDMINSTER, NJ 07921

THIRD PARTY REQUESTER:

VANABLE LLP
1290 AVENUE OF THE AMERICAS
NEW YORK, NY 10104-3800

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re *Ex Parte* Reexamination of:	Confirmation No. Not yet assigned
U.S. Patent No.: 8,327,011	Reexam Control No.: Not yet assigned
Issue Date: December 4, 2012	Reexam Group AU: Not yet assigned
Application No. 13/374,942	
Filing Date: January 24, 2012	
Inventor: Harold Edward Price	
Title: STREAMING MEDIA BUFFERING SYSTEM	26694 PATENT & TRADEMARK OFFICE

REQUEST FOR *EX PARTE* REEXAMINATION OF U.S. PATENT NO. 8,327,011

via EFS-Web
Mail Stop *Ex Parte* REEXAM
Attn: Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Commissioner:

FriendFinder Networks Inc. and Streamray Inc. ("Requesters") respectfully request *ex parte* reexamination under 35 U.S.C. §§ 301 *et seq*. and 37 C.F.R. § 1.510 *et seq*. of claims 1 and 4 of U.S. Patent No. 8,327,011 (the "'011 Patent"), issued on December 4, 2012 to Harold Edward Price and titled, "Streaming Media Buffering System," a single-sided copy of which is attached to this Request as Exhibit 2. WAG Acquisition, LLC ("WAG") claims to be the assignee of the '011 Patent.

The term of the '011 Patent has expired.

The reexamination fee of $12,600.00 under 37 C.F.R. § 1.20(c)(2) is included with this Request.

Pursuant to 37 C.F.R. § 1.510, this Request for *Ex Parte* Reexamination includes:

1. A statement pointing out each substantial new question of patentability based on prior art patents and printed publications;
2. An identification of every claim for which reexamination is requested and a detailed explanation of the pertinency and manner of applying the cited prior art to every claim for which reexamination is requested;
3. A copy of every patent or printed publication relied upon or referred to herein, attached as Exhibits 1-18;
4. A copy of the entire '011 Patent for which reexamination is requested, including any disclaimer, certificate of correction, or reexamination certificate issued in the patent;
5. A certification that a copy of the Request filed by Requesters has been served in its entirety on the patent owner at the address as provided for in 37 C.F.R. § 1.33(c) (filed separately);
6. A certification that the statutory estoppel provisions of 35 U.S.C. § 315(e)(1) or 35 U. S.C. § 325(e)(1) do not prohibit the Requesters from filing the *Ex Parte* Reexamination Request; and
7. The appropriate fee under 37 C.F.R. § 1.20(c). The Commissioner is authorized to charge any additional fees, or to credit any overpayment, to Deposit Account No. 22-0261.

TABLE OF CONTENTS

TABLE OF EXHIBITS[1]

Exhibit No.	Description	Shorthand
Copies of Patents Being Challenged Under Reexamination		
1	U.S. Patent No. 8,122,141	’141 Patent
2	U.S. Patent No. 8,327,011	’011 Patent
Prior Art References		
3	U.S. Patent No. 6,389,473 to Carmel et al.	Carmel
4	U.S. Patent No. 7,529,806 to Shteyn	Shteyn
5	WO1997044942 to Kliger et al.	Kliger
6	U.S. Patent No. 6,175,862 to Chen and Feig	Feig
7	U.S. Patent No. 5,987,510 to Imai et al.	Imai
8	U.S. Patent No. 6,792,468 to Bloch et al.	Bloch
9	U.S. Patent No. 5,793,980 to Glaser et al.	Glaser
10	U.S. Patent No. 6,005,600 to Hill	Hill
11	U.S. Patent No. 6,845,398 to Galensky et al.	Galensky
File Histories and IPR/Litigation Materials		
12	Final Written Decision on Remand, *WebPower v. WAG Acquisition*, IPR2016-01238, Paper No. 28 (Jul. 16, 2020)	WebPower Final Written Decision on Remand
13	’141 Patent Prosecution History	’141 Patent Pros. History

[1] Requesters also intend to file a request for *Ex Parte* Reexamination on U.S. Patent Nos. 8,364,839 to Price; 8,185,611 to Price; and 8,122,141 to Price. To facilitate the Office’s review, Requesters have used the same exhibit numbering across the ’141 and ’011 Requests.

Exhibit No.	Description	Shorthand
14	’011 Patent Prosecution History	’011 Patent Pros. History
15	Plaintiff's Opening Claim Construction Brief, *WAG Acquisition, L.L.C. v. Gattyàn Group S.à r.l. et al*, No. 2:14-cv-2832-ES-MAH, Dkt. No. 274 (D.N.J. Jan. 7, 2021)	WAG-NJ Op. CC Br.
16	Plaintiff's Responsive Claim Construction Brief, *WAG Acquisition, L.L.C. v. Gattyàn Group S.à r.l. et al*, No. 2:14-cv-2832-ES-MAH, Dkt. No. 282 (D.N.J. Mar. 4, 2021)	WAG-NJ Resp. CC Br.
17	Plaintiff's Opening Claim Construction Brief, *WAG Acquisition, L.L.C. v. Flying Crocodile, Inc. et al.*, No. 2:19-cv-01278-BJR, Dkt. No. 263 (W.D. Wash. Mar. 24, 2021)	WAG-Wash Op. CC Br.
18	Plaintiff's Responsive Claim Construction Brief, *WAG Acquisition, L.L.C. v. Flying Crocodile, Inc. et al.*, No. 2:19-cv-01278-BJR, Dkt. No. 269 (W.D. Wash. Apr. 14, 2021)	WAG-Wash Resp. CC Br.

I. IDENTIFICATION OF CLAIMS FOR WHICH REEXAMINATION IS REQUESTED

In accordance with 37 C.F.R. § 1.510, reexamination of claims 1 and 4 of '011 Patent is requested in view of the following references:

- U.S. Patent No. 7,529,806 to Shteyn ("Shteyn") (Ex. 4)
- U.S. Patent No. 6,005,600 to Hill ("Hill") (Ex. 10)
- U.S. Patent No. 6,175,862 to Feig et al. ("Feig") (Ex. 6)
- U.S. Patent No. 6,389,473 to Carmel et al. ("Carmel") (Ex. 3)
- U.S. Patent No. 6,792,468 to Bloch et al. ("Bloch") (Ex. 8)
- WO1997044942 to Kliger et al. ("Kliger") (Ex. 5)
- U.S. Patent No. 5,987,510 to Imai et al. ("Imai") (Ex. 7)
- U.S. Patent No. 5,793,980 to Glaser et al. ("Glaser") (Ex. 9)

Such claims are presently asserted in a litigation against Requesters in *WAG Acquisition, L.L.C. v. FriendFinder Networks Inc., et al.*, No. 19-cv-05036-JD and separately against another party. *See* Table 1, No. 1 below.

Reexamination of claims 1 and 4 is requested under 35 U.S.C. §§ 102, 103(a) as being unpatentable over:

1. Shteyn
2. Shteyn and Carmel, Glaser, or TCP
3. Hill or Hill and Carmel, Glaser, or TCP
4. Shteyn and Hill
5. Feig or Feig and Carmel, Glaser, or TCP
6. Carmel or Carmel and Hill
7. Bloch or Bloch and Carmel, Glaser, or TCP
8. Kliger
9. Imai and Carmel, Glaser, or TCP

Pursuant to 37 C.F.R. § 1.510, *ex parte* reexamination is available "at any time during the period of enforceability of a patent."

The ’011 Patent claims priority on its front page to a continuation of application no. 12/800,152, filed on May. 10, 2010, now U.S. Patent No. 8,122,141, which is a continuation of application no. 10/893,814, filed on Jul. 19, 2004, now U.S. Patent No. 7,716,358, which is a continuation-in-part of application no. 09/819,337, filed on Mar. 28, 2001, now U.S. Patent No. 6,766,376, and a provisional application (60/231,997) filed on September 12, 2000. *See* ’011 Patent (Ex. 2), at front page. Lastly, there is no adjustment of patent term under 35 U.S.C. § 154(b) listed on the front page of the ’011 Patent. ’011 Patent (Ex. 2), at front page. Thus, the ’011 Patent expired on March 28, 2021.

Accordingly, the ’011 Patent is eligible for *ex parte* reexamination because it is within the six-year window for *ex parte* reexamination after its expiration. *See* MPEP § 2211 (explaining that, in assessing eligibility for *ex parte* reexamination, “the period of enforceability is generally determined by adding 6 years to the date on which the patent expires”).

II. <u>OVERVIEW OF U.S. PATENT NO. 8,327,011</u>

A. Description of the Alleged Invention of the ’011 Patent

The ’011 Patent is directed to methods and systems for sending streaming media, such as audio or video files, via the Internet. ’011 Patent (Ex. 2), at 4:28-30. The ’011 Patent admits that sending audio and video files via a network was known in the art. *Id.* at 3:56-57. The ’011 Patent also admits that it was known for media data stored in a server to be sent over networks to a client buffer to assure a continuous stream of video. *Id.* at 2:24-29. The ’011 Patent further admits that it was known to use a pre-buffering technique so that the video can be played with a minimum of dropouts, and that it was known to transmit video at the rate it is to be played back on the associated media player. *Id.* at 2:45-52. Indeed, the ’011 presumes a transport mechanism “for the reliable delivery of data.”

> This invention presumes the existence of a data communications transport mechanism, such as the TCP protocol, for the reliable delivery of data in an ordered sequence from the source of the media data to the server, or from the server to the media player software of the user computer. Thus, the delivery of data in the proper sequence is outside the scope of this invention.

Id. at 4:61-67. The Applicant further conceded that the purported invention "sends" or transmits data faster than the playback rate by virtue of known TCP:

> In one method of operation, the resending of missing data is the responsibility of a **reliable transport mechanism, such as TCP**. The server buffer "sends" data by delivering it to the transport mechanism. The **transport mechanism actually "sends" the data across the communications medium**, and has processes which determine if all the data that has been sent has been received by the destination. If not, missing pieces of data are automatically resent to the destination, and are arranged to be delivered to the target software on the destination system in an ordered fashion. In the circumstance of this invention, the destination is the user computer, and the target software on the destination system is the media player. If the **transport mechanism determines that data is missing, it retransmits that data to the destination as fast as the connection between the server and destination will allow**. The net effect of this invention is that all media data to be delivered to a user computer is always **sent as fast as the communications medium will support**, **either by the server buffer passing media data to the transport mechanism, or by the transport mechanism delivering or redelivering the media data to the user computer**.

Id. at 5:63-6:16 (emphasis added).

Against this backdrop of well-known techniques, the '011 Patent asserts that there was a need for improved systems and methods to "afford immediate and uninterrupted listening/viewing of streaming media by the user." *Id.* at 4:22-24. But this was known in the art too, as shown in the below prior art references.

B. Summary of the Prosecution History of the '011 Patent

During prosecution of the '011 Patent, no office action was issued by the United States Patent & Trademark Office ("USPTO"). *See* '011 Patent Pros. History (Ex. 14).

C. Priority Date of the '011 Patent

The '011 Patent claims priority on its front page to a continuation of application no. 12/800,152, filed on May. 10, 2010, now U.S. Patent No. 8,122,141, which is a continuation of application no. 10/893,814, filed on Jul. 19, 2004, now U.S. Patent No. 7,716,358, which is a continuation-in-part of application no. 09/819,337, filed on Mar. 28, 2001, now U.S. Patent No. 6,766,376, and a provisional application (60/231,997) filed on September 12, 2000. *See* '011 Patent (Ex. 2), at front page. For purposes of this Request only, Requesters apply a priority date of September 12, 2000.

III. LEVEL OF SKILL IN THE ART

A person of ordinary skill in the art ("POSITA") for the '011 Patent would have a Bachelor of Science degree in computer science or electrical engineering (or comparable degree) and two years of experience in networking or streaming media, or a Master of Science in computer science or electrical engineering (or comparable degree). The Patent Trial and Appeal Board ("PTAB") previously adopted this level in an *Inter Partes* Review ("IPR"), and Requesters believe for the purposes of this Request that such level should be adopted here. WebPower Final Written Decision on Remand (Ex. 12), at 12-13.

IV. NOTIFICATION OF CONCURRENT PROCEEDINGS

A. Prior '011 Patent IPR Proceedings

Requesters, Duodecad IT Services Luxembourg S.à r.l., and others previously filed an IPR on the '011 Patent, but it was denied institution because the PTAB determined that a reference (Su) did not qualify as a printed publication or prior art. Decision Denying Institution of *Inter Partes* Review, *FriendFinder Networks Inc. et al. v. WAG Acquisition L.L.C.*, IPR2015-01033,

Paper No. 8, at 6-14 (Oct. 19, 2015). Su is not relied on here and thus that PTAB decision respectfully has no bearing on this Request.

Additionally, WebPower, Inc. previously filed an IPR on the '011 Patent, but it was denied institution based on prior art references (Chen; Chen and Rolf). Decision Denying Institution of *Inter Partes* Review, *WebPower v. WAG Acquisition L.L.C.*, IPR2016-01161, Paper No. 7, at 8, 12-21 (Dec. 12, 2016). Chen (or Rolf) is not relied on here and thus that PTAB decision respectfully has no bearing on this Request.

Lastly, I.M.L. SLU previously filed an IPR on the '011 Patent, but it was denied institution based on prior art references (Chen; Chen and Chen File History; and Chen, Galensky, and White). Decision Denying Institution of *Inter Partes* Review, *I.M.L. SLU v. WAG Acquisition L.L.C.*, IPR2016-01655, Paper No. 11, at 8, 10-15 (Feb. 27, 2017). Chen as a primary reference is not relied on here and thus that PTAB decision respectfully has no bearing on this Request.

Accordingly, Requesters believe (as explained below) that this Request raises a substantial new question of patentability on claims 1 and 4 on multiple grounds. This is particularly true given the broad claim constructions WAG has recently proposed in litigation as discussed below.

B. List of Current and Prior Proceedings

The '011 Patent is currently the subject of pending litigation and related *ex parte* reexaminations (Table 1 below) as well as prior litigations and *Inter Partes* review ("IPR") (Table 2 below).

<u>Table 1: Current Litigations and *Ex Parte* Reexaminations</u>

No.	Case Caption	Court	Patents	Status
1	*WAG Acquisition, L.L.C. v. Flying Crocodile, Inc. et al.*, No. 2:19-cv-01278-BJR	WDWA (transferred from DNJ)	8,122,141 8,327,011 8,185,611 8,364,839	*Markman* briefing is under submission as of April 14, 2021. Dkt. Nos. 262-270.
2	*WAG Acquisition, L.L.C. v. FriendFinder Networks Inc., et al.*, No. 19-cv-05036-JD	NDCA (transferred from DNJ)	8,122,141 8,327,011 8,185,611 8,364,839	Requesters are Defendants in this Litigation. Status report filed on July 30, 2020. Dkt. No. 225.
3	8,122,141 *Ex Parte* Reexamination	USPTO	8,122,141	Requesters filed on August 25, 2021
4	8,327,011 *Ex Parte* Reexamination	USPTO	8,327,011	Requesters filed on August 25, 2021
5	8,185,611 *Ex Parte* Reexamination	USPTO	8,185,611	Requesters filed on August 25, 2021
6	8,364,839 *Ex Parte* Reexamination	USPTO	8,364,839	Requesters filed on August 25, 2021

<u>Table 2: Prior Litigations and IPRs[2]</u>

No.	Case Caption	Court	Patent(s)	Status
1	*WAG Acquisition, L.L.C. v. Gattyàn Group S.à r.l. et al.*, No. 2:14-cv-2832-ES-MAH	DNJ	8,122,141 8,327,011 8,185,611 8,364,839	The Parties had completed claim construction or *Markman* briefing. Dkt. Nos. 281-282.

[2] For purposes of inclusiveness, this Table 2 provide prior resolved litigations and IPRs for patents in the same family as the '011 Patent.

				Parties stated on August 23, 2021: "we write to advise the Court that the Parties have a fully-executed settlement agreement, and will be filing a stipulation of dismissal shortly." Dkt. No. 306
2	*WAG Acquisition, L.L.C. v. Sobonito Investments, Ltd. et al.*, No. 2:14-cv-1661-ES-MAH	DNJ	8,122,141 8,327,011 8,185,611 8,364,839	Dismissed with prejudice So Ordered on September 8, 2020. Dkt. No. 213.
3	*WAG Acquisition, L.L.C. v. Vubeology, Inc. et al.*, No. 1:19-cv-00805-LY	WDTX (transferred from DNJ)	8,122,141 8,327,011	Dismissed without prejudice So Ordered on April 19, 2021. Dkt. No. 174.
4	*WAG Acquisition L.L.C. v. WebPower, Inc. et al.*, No. 9:19-cv-81155-RAR	SDFL (transferred from DNJ)	8,122,141 8,327,011 8,364,839 8,185,611	Dismissed with prejudice So Ordered on February 7, 2021. Dkt. No. 191.
5	*WAG Acquisition, L.L.C. v. Data Conversions, Inc. et al.*, No. 3:19-cv-00489-MMD-CBC	DNV (transferred from DNJ)	8,122,141 8,327,011	Dismissed with prejudice So Ordered on March 17, 2021. Dkt. No. 278.
6	*WAG Acquisition, LLC v. Multi Media, L.L.C. et al.*, No. 2:19-cv-07076-SJO-GJS	CDCA (transferred from DNJ)	8,122,141 8,327,011	Dismissed with prejudice So Ordered on January 8, 2021. Dkt. No. 266.

7	*WAG Acquisition, L.L.C. v. XM Satellite Radio, Inc.* et al, No. 1:08-cv-06357-RMB-MHD	SDNY	6,766,376	Stipulation of dismissal So Ordered on February 9, 2009. Dkt. No. 33.
8	*WAG Acquisition, L.L.C. v. MFCXY, Inc. et al.*, No. 2:14-cv-3196-ES-MAH	DNJ	8,122,141 8,327,011 8,364,839 8,185,611	Stipulation of dismissal So Ordered on May 8, 2015. Dkt. No. 48.
9	*WAG Acquisition, L.L.C. v. Gamelink Int'l Ltd. et al.*, No. 2:15-cv-3416-ES-MAH	DNJ	8,122,141 8,327,011	Notice of dismissal So Ordered on November 8, 2018. Dkt. No. 56.
10	*WebPower et al. v. WAG Acquisition LLC*, IPR2016-01238[3]	PTAB	8,122,141	The PTAB found on July 16, 2020 (Paper No. 28) claims 10–18 unpatentable in a final written decision, which was not appealed by WAG. Previously claims 19-23 were found invalid and not appealed. *WAG Acquisition, LLC v. WebPower*, Inc., 781 F. App'x 1007, 1009 n.2 (Fed. Cir. 2019) ("The Board

[3] IPR2017-00820 and IPR2017-00786 were joined with IPR2016-01238.

				also instituted review of claims 19-23 on multiple grounds. WAG has not challenged the Board's unpatentability determinations with respect to these claims.")
11	*WebPower et al. v. WAG Acquisition L.L.C.*, IPR2016-01239[4]	PTAB	8,364,839	On December 26, 2017 (Paper No. 21), the PTAB found claims 5, 12, and 19 unpatentable in a final written decision. On August 26, 2019, the Federal Circuit issued a Fed. Cir. R. 36 affirmance. *See WAG Acquisition, LLC v. WebPower, Inc.*, 2019 U.S. App. LEXIS 25497, at *1 (Fed. Cir. Aug. 26, 2019).
12	*Duodecad IT Services Luxembourg S.à r.l. et al v. WAG Acquisition L.L.C.*, IPR2015-01036	PTAB	8,364,839	On October 20, 2016 (Paper No. 17), the PTAB found claims 1, 3, 4, 6, 8, 10, 11, 13, 15, 17, 18,

[4] IPR2017-00784 and IPR2017-00785 were joined with IPR2016-01239.

				and 20 unpatentable in a final written decision, which was not appealed by WAG.
13	*Duodecad IT Services Luxembourg S.à r.l. et al v. WAG Acquisition L.L.C.*, IPR2015-01035	PTAB	8,185,611	Denied institution.
14	*FriendFinder Networks Inc. et al. v. WAG Acquisition L.L.C.*, IPR2015-01037	PTAB	8,122,141	Denied institution.
15	*FriendFinder Networks Inc. et al. v. WAG Acquisition L.L.C.*, IPR2015-01033	PTAB	8,327,011	Denied institution.
16	*WebPower v. WAG Acquisition L.L.C.*, IPR2016-01161	PTAB	8,327,011	Denied institution.
17	*WebPower v. WAG Acquisition L.L.C.*, IPR2016-01162	PTAB	8,185,611	Denied institution.
18	*I.M.L. SLU v. WAG Acquisition L.L.C.*, IPR2016-01655	PTAB	8,327,011	Denied institution.
19	*I.M.L. SLU v. WAG Acquisition L.L.C.*, IPR2016-01656	PTAB	8,122,141	Terminated.
10	*I.M.L. SLU v. WAG Acquisition L.L.C.*, IPR2016-01657	PTAB	8,185,611	Denied institution.
21	*I.M.L. SLU v. WAG Acquisition L.L.C.*, IPR2016-01658[5]	PTAB	8,364,839	Terminated (Paper No. 46). Also, IPR2017-01179 joinder was vacated.

In view of the above concurrent litigation (see Table 1), it is respectfully urged that this Request be granted and reexamination be conducted not only with "special dispatch," but also with "priority over all other cases" in accordance with M.P.E.P. § 2261.

V. <u>RELATED REEXAMINATION PROCEEDINGS</u>

Requesters are aware of the reexamination proceedings as indicated above in Table 1.

---

[5] IPR2017-01179 was joined with IPR2016-01658.

VI. STATEMENT POINTING OUT EACH SUBSTANTIAL NEW QUESTION OF PATENTABILITY

A. Background on the Prior Art forming the SNQPs

1. Shteyn is Analogous Prior Art

Shteyn was filed on November 4, 1999 and issued on May 5, 2009. It is prior art to the ’011 Patent under at least (pre-AIA) 35 U.S.C. § 102(e). Shteyn is not cited on the front page of the ’011 Patent.

Shteyn teaches a computer with media player software, including Java, for playing streaming media. Shteyn (Ex. 4), at 4:32-44. Shteyn further teaches that the client downloads a control file listing segments each with a serial number, *e.g.*, URL or filename, available at the server. *Id.* at 2:60-63, 2:67-3:2. Shteyn teaches that such segments can have different bandwidth, and the client can select the appropriate bandwidth segment depending on network bandwidth. *Id.* at 3:44-53, 3:55-56, 4:20-26, Fig. 2, claims 1, 12.

Shteyn describes that the media player buffers and plays the first segment (*e.g.*, part1) while downloading, buffering, and playing the second segment (*e.g.*, part2) and other segments until the end of the video or program. *Id.* at Abstract, 3:3-18. Shteyn further teaches that the media player can implement this buffering using a linked list of segments (data structure that records the name of the segment and data with a pointer to the next segment received), which tracks the last segment with serial number downloaded and prior segments. *Id.* at 3:14-18, 3:31-36.

2. Hill is Analogous Prior Art

Hill was filed on October 18, 1996 and issued on December 21, 1999. It is prior art to the ’011 Patent under at least (pre-AIA) 35 U.S.C. §§ 102 (a) and (e). Hill is not cited on the front page of the ’011 Patent.

Hill describes a media player receiving a sequence of frames (which form a clip(s)) over the Internet and playing them. Hill (Ex. 10), at 1:7-11, 3:27-36, 3:46-56. Each frame is associated with a frame or serial number, and the media player repeatedly requests and downloads one frame or two frames by the frame number. *Id.* at 5:52-54, 5:63-6:16, 7:12-24, 11:63-12:13. Hill further describes a player buffer manager that maintains a threshold of frames and a prior history of downloaded frames. *Id.* at 6:58-67, 7:1-2, 9:40-44, 10:55-60, claims 8-9. Hill also teaches that the predetermined frames can be sent at 48 frames per second while playing back the frames at 24 frames per second. *Id.* at 7:12-24.

3. Feig is Analogous Prior Art

Feig was filed on June 17, 1998 and issued on January 16, 2001. It is prior art to the '011 Patent under at least (pre-AIA) 35 U.S.C. § 102(e). Feig is not cited on the front page of the '011 Patent.

Feig teaches a media player that downloads from a server a file with a sequence of URLs, where each URL is unique or serial number that represents a video segment. Feig (Ex. 6), at 1:56-57, 2:54-59, 2:60-67. Using the file with the listing of URLs, the media player requests each media segment by URL or serial number. *Id.* at Abstract, 5:16-23. Feig further teaches that the media player uses two buffers to repeatedly store and play segments until the end of the program. *Id.* at 5:16-23, 5:37-43, claim 11. Feig provides an example of transmitting such segments at a rate faster than the playback rate and buffering as well as playing such segments. *Id.* at 6:21-38.

4. Carmel is Analogous Prior Art

Carmel was filed on March 24, 1999 and issued on May 14, 2002. It is prior art to the '011 Patent under at least (pre-AIA) 35 U.S.C. § 102 (e). Carmel is cited on the front page of the '011 Patent.

Carmel discloses a method for real-time media streaming from a server to a plurality of client computers over the Internet. Carmel (Ex. 3), at 2:1-21. Carmel discloses dividing the media data into sequential slices (or segments), each having a unique index or serial number. *Id.* at Fig. 3A, 7:18-35. Carmel further teaches that a server provides an index file consisting of these slices with a serial number. *Id.* at 7:59-8:5. Carmel also teaches that the index file at the server can consist of different quality level or resolution files. *Id.* at 3:5-9, 8:42-9:5, Fig. 3D.

The client reads the server's index file containing the numbered slices and thereafter repeatedly requests and downloads the numbered slices from the server. *Id.* at 7:59-8:5, 10:25-48, claim 10, Fig. 6A. The downloaded slices with serial number are decoded and played to the client (*id.* at 7:4-17, 10:25-48, 10:64-11:8, Figs. 6A-6B) as well as can be displayed to the client in the form of a graphical slider. *Id.* at Fig. 3C, 8:18-31.

5. **Bloch is Analogous Prior Art**

Bloch was filed on November 16, 1999 and issued on September 14, 2004. It is prior art to the '011 Patent under at least (pre-AIA) 35 U.S.C. § 102(e). Bloch is not cited on the front page of the '011 Patent.

Bloch teaches that a media player with software (playback engine) requests frames of a clip(s) by frame or serial number. Bloch (Ex. 8), at Abstract, 4:57-61, 5:40-46, 5:52-59, 6:63-7:3, 9:12-22. In response, the server transmits the requested frame numbers for the clips to the media player for buffering and playing the entire clip(s). *Id.* at 7:30-8:4, 9:41-46, Fig. 6. Bloch also provides to the user a graphical user interface which tracks and displays the frame numbers received. *Id.* at 5:60-6:7, Fig. 7.

6. **Kliger is Analogous Prior Art**

Kliger has an international filing date of May 22, 1997 and published on November 27, 1997. It is prior art to the ’011 Patent under at least (pre-AIA) 35 U.S.C. §§ 102(a) and (b). Kliger is not cited on the front page of the ’011 Patent.

Kliger teaches that a media player with software requests from the server an object map with a listing of objects (*e.g.*, video) by serial number. Kliger (Ex. 5), at 7:10-12, 8:3-15, 8:16-8:25, 9:20-9:29. Kliger further teaches that the client application uses the object map to request the objects in the map by serial number. *Id.* at Abstract, 3:8-11, 8:3-15, Fig. 3. Kliger also describes that the objects can have different bandwidth, where the client can select the appropriate bandwidth object depending on network bandwidth. *Id.* at 3:25-32, 5:25-28, 9:32-10:4, 12:33:13:8, claim 3.

Kliger’s client application will buffer each object associated with a serial number received from the server (*id.* at 11:34-12:5) and will manage or monitor its buffers to ensure the transmission rate is greater than the consumption (or playback) rate. *Id.* at 13:33-14:2, 15:10-13, 18:21-19:2.

7. **Imai is Analogous Prior Art**

Imai was filed on November 8, 1996 and issued on November 16, 1999. It is prior art to the ’011 Patent under at least (pre-AIA) 35 U.S.C. §§ 102(a) and (e). Imai is not cited on the front page of the ’011 Patent.

Imai teaches that a media player or Java program downloads a file list with a listing of files available for download from the server. Imai (Ex. 7), at Abstract, 1:28-31, 20:58-64, 20:65-21:4. After downloading that file list, the client issues request for file(s) by serial number (URL or filename) to the server and stores each file downloaded from that file list by serial number. *Id.* at 22:11-33, 13:1-13.

8. Glaser is Analogous Prior Art

Glaser was filed on November 30, 1994 and issued on August 11, 1998. It is prior art to the ’011 Patent under at least (pre-AIA) 35 U.S.C. §§ 102(a), (b) and (e). Glaser is cited on the front page of the ’011 Patent.

Glaser teaches a streaming media system in which data is transmitted over the Internet using TCP/IP to clients. Glaser (Ex. 9), at Abstract, 7:1-28, 21:57-63. Glaser describes maintaining the client buffers at or near maximum capacity to guard against dropouts. *Id.* at 3:15-36, 13:36-44. To maintain this level, the client requests the server to send normal quality data at a rate faster than the playback rate. *Id.* at 16:2-10, 20:8-16, 21:57-63, 22:41-59.

B. The Grounds for the SNQPs

“A prior art patent or printed publication raises a substantial question of patentability where there is a substantial likelihood that a reasonable examiner would consider the prior art patent or printed publication important in deciding whether or not the claim is patentable.” MPEP § 2242.

Requesters respectfully submit that a substantial new question of patentability exists with respect to claims 1 and 4 under 35 U.S.C. § 102 as anticipated by Shteyn.

Requesters respectfully submit that a substantial new question of patentability exists for claims 1 and 4 under 35 U.S.C. § 103 as obvious over Shteyn and Carmel, Glaser, or TCP.

Requesters respectfully submit that a substantial new question of patentability exists with respect to claims 1 and 4 under 35 U.S.C. § 102 as anticipated by Hill or under 35 U.S.C. § 103 as obvious over Hill and Carmel, Glaser, or TCP.

Requesters respectfully submit that a substantial new question of patentability exists with respect to claims 1 and 4 under 35 U.S.C. § 103 as obvious over Shteyn and Hill.

Requesters respectfully submit that a substantial new question of patentability exists with respect to claims 1 and 4 under 35 U.S.C. § 102 as anticipated by Feig or under 35 U.S.C. § 103 as obvious over Feig and Carmel, Glaser, or TCP.

Requesters respectfully submit that a substantial new question of patentability exists with respect to claims 1 and 4 under 35 U.S.C. § 102 as anticipated by Carmel or under 35 U.S.C. § 103 as obvious over Carmel and Hill.

Requesters respectfully submit that a substantial new question of patentability exists with respect to claims 1 and 4 under 35 U.S.C. § 102 as anticipated by Bloch or under 35 U.S.C. § 103 as obvious over Bloch and Carmel, Glaser, or TCP.

Requesters respectfully submit that a substantial new question of patentability exists with respect to claims 1 and 4 under 35 U.S.C. § 102 as anticipated by Kliger.

Requesters respectfully submit that a substantial new question of patentability exists with respect to claims 1 and 4 under 35 U.S.C. § 103 as obvious over Imai and Carmel, Glaser, or TCP.

VII. <u>CLAIM CONSTRUCTION</u>

A. Legal Standard

For the purposes of this Request, the claim terms are presented by Requesters in accordance with M.P.E.P. §§ 2258.I.G. for an expired patent.

In a reexamination proceeding involving claims of an expired patent, claim construction pursuant to the principles set forth in *Phillips v. AWH Corp.*, 415 F.3d 1303, 1316 (Fed. Cir. 2005) should be applied because the expired claims are not subject to amendment. M.P.E.P. §§ 2258.I.G ("In a reexamination proceeding involving claims of an expired patent, claim construction pursuant to the principle set forth by the court in *Phillips* . . . should be applied since the expired claim are not subject to amendment.") (citation omitted); *Collabo Innovations, Inc. v. Sony Corp.*, 778 F.

App'x 954, 957 (Fed. Cir. 2019) ("Because the '714 patent has expired, the claim construction standard set forth in *Phillips* . . . applies.") (citation omitted) (unpublished); *Veraseal LLC v. Wal-Mart Stores, Inc.*, No. 2:17-CV-00527-JRG, 2018 U.S. Dist. LEXIS 79479, at *5 (E.D. Tex. May 10, 2018) (noting during reexamination that claims of an expired patent are construed according to *Phillips*).

A *Phillips* claim construction means words are "generally given their ordinary and customary meaning . . . the term would have to a person of ordinary skill in the art in question at the time of the invention . . . ." *Phillips*, 415 F.3d at 1312-13 (citations and quotations omitted). Such plain and ordinary meaning construction is based on the intrinsic record (patent and prosecution history). *Id.*; *Eon Corp. IP Holdings LLC v. Silver Spring Networks, Inc.*, 815 F.3d 1314, 1320 (Fed. Cir. 2016) ("The ordinary meaning of a claim term is not the meaning of the term in the abstract. Instead, the ordinary meaning of a claim term is its meaning to the ordinary artisan after reading the entire patent.") (citations and quotations omitted). Dictionaries and other extrinsic evidence are relevant but "less significant than the intrinsic record in determining the legally operative meaning of claim language." *Phillips*, 415 F.3d at 1317 (quotations and citations omitted).

B. Summary of Prior Claim Constructions

No district court has issued a claim construction decision on the '011 Patent. However, as indicated above, WAG has filed in 2021 claim construction briefs in district court litigations involving other parties. *See* Table 1, No. 1; Table 2, No. 1. This includes WAG proposing broad constructions for terms in the '011 Patent. *See* Exs. 15-18.

As such, Requesters believe the terms of claims 1 and 4 of the '011 Patent should be construed no narrower than the broad plain and ordinary meaning constructions set forth by WAG

under *Phillips* in attempting to enforce the '011 Patent in district court litigations. This ensures "that claims are not argued one way in order to maintain their patentability and in a different way against accused infringers," especially here where the district court litigations and the *ex parte* reexam are both analyzed under the *Phillips* claim construction standard. *Aylus Networks, Inc. v. Apple Inc.*, 856 F.3d 1353, 1360 (Fed. Cir. 2017); *Commscope Techs. LLC v. Dali Wireless*, Nos. 2020-1817, 2020-1818, 2021 U.S. App. LEXIS 25294, at *16-17 (Fed. Cir. Aug. 24, 2021) (improper for patentee to "simultaneously argue" inconsistent infringement and validity positions); *Amazon.com, Inc. v. Barnesandnoble.com, Inc.*, 239 F.3d 1343, 1351 (Fed. Cir. 2001) ("the claims must be interpreted and given the same meaning for purposes of both validity and infringement analyses.") (internal citation omitted); *Sterner Lighting, Inc. v. Allied Elec. Supply, Inc.*, 431 F.2d 539, 544 (5th Cir. 1970) (citing *White v. Dunbar*, 119 U.S. 47, 51, 30 L. Ed. 303, 7 S. Ct. 72 (1886)) ("A patent may not, like a 'nose of wax,' be twisted one way to avoid anticipation and another to find infringement.") It is especially true for the broad plain and ordinary meaning constructions WAG has proposed under *Phillips* in litigation for claim limitations 1[a] (serial number / identifier) and 1[g] (predetermined).

WAG's Serial Number/Identifier (1[a]). WAG contends that the plain and ordinary meaning under *Phillips* of "serial number / identifier" is **not limited** to a "consecutive [numbers] or limited to numerals." WAG-NJ Resp. CC Br. (Ex. 16), at 19 ("Nothing in the plain language of these two terms requires them to be either consecutive or limited to numerals, as argued in WAG's opening brief. See D.I. 274 at 22-23.") Thus, under WAG's broad understanding, a serial number/identifier can broadly encompass several forms.

WAG's Predetermined (1[g]). WAG contends under *Phillips* that the claim language "to repeat transmitting the requests to the media source for sequential media data elements so as to

maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received" has a broad plain and ordinary meaning. Specifically, during litigation WAG asserted that "the Court should give it its plain and ordinary meaning: to maintain a **specified number of media data elements** in the player buffer until the last media data element comprising the program has been received." WAG-NJ Op. CC Br. (Ex. 15), at 18-19 (emphasis added); *see also* Decision Denying Institution of *Inter Partes* Review, *WebPower v. WAG Acquisition L.L.C.*, IPR2016-01161, Paper No. 7, at 8-10 (Dec. 12, 2016).

WAG further contented that the predetermined number is not "constant" or the "same number" and could be determined at any point (before or after connecting to the server). WAG-NJ Resp. CC Br. (Ex. 16), at 16 ("As for its justification for rewriting 'the pre-determined number of media data elements' to read as 'the same number of media data elements,' all Duodecad can point to is an excerpt from the '011 patent that discusses an embodiment in which the buffer level remains constant . . . . But Duodecad offers no reason why language concerning an embodiment in the specification must be imported into the claims . . . .") (internal citations omitted); WAG-Wash Resp. CC Br. (Ex. 18), at 7 (same); WAG-Wash Op. CC Br. (Ex. 17), at 16-17 ("But there is no reason for this broad term to be narrowly construed to a single criteria of being set 'prior to connecting to the server,' as FCI would have it. . . . [A] POSITA would understand that the 'predetermined' number of media data elements could be determined after connection to the network and after connection to the media source.") (citation and quotations omitted)

Thus, under WAG's "broad" proposed plain and ordinary meaning construction, a prior art reference teaching any specified data in the client buffer at any point (whether before the client connects to the server or after the client connects to the server) would cover this claim limitation.

Rate. Based on the WebPower IPR discussed above, the Federal Circuit construed "rate" in the '141 patent as "the rate at which each requested media data element is transmitted from the server to the user computer. . . . The rate limitation in claim 10 therefore refers to the rate at which requested media data elements are sent, not the overall rate at which data is transmitted from the server to the user computer." *WAG Acquisition, LLC v. WebPower, Inc.*, 781 F. App'x 1007, 1011-1012 (Fed. Cir. 2019); WebPower Final Written Decision on Remand (Ex. 12), at 9-10.

Other than the above constructions, including the Federal Circuit's construction, Requesters believe for the purposes of this Request only that the terms of claims 1 and 4 of the '011 Patent should take on their ordinary and customary meaning that the terms would have to a POSITA.

The prior art teaches the challenged claims 1 and 4, particularly under WAG's broad plain and ordinary meaning constructions.

VIII. DETAILED EXPLAINATION UNDER 37 C.F.R. § 1.510(B)(2)

"A prior art document may anticipate a claim if it describes every element of the claimed invention, either expressly or inherently." *Nobel Biocare Servs. AG v. Instradent USA, Inc.*, 903 F.3d 1365, 1375 (Fed. Cir. 2018) (internal citation omitted).

A claim is obvious when "the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). This obviousness analysis involves determining "the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved." *Id.* (quoting *Graham v. John Deere Co.*, 383 U.S. 1, 17 (1966)). A

claimed combination is obvious if there was an apparent reason for a skilled artisan "to combine the known elements in the fashion claimed by the patent at issue." *Id.* at 418.

There are numerous reasons that a POSITA would combine known elements or modify existing prior art.

> [T]he Court [in *KSR*] explained that a rationale to combine could arise from interrelated teachings of multiple patents; the effects of demands known to the design community or present in the market-place; and the background knowledge possessed by a person having ordinary skill in the art. For example, the Court stated that if a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill. For the technique's use to be obvious, the skilled artisan need only be able to recognize, based on her background knowledge, its potential to improve the device and be able to apply the technique.

Unwired Planet, LLC v. Google Inc., 841 F.3d 995, 1003 (Fed. Cir. 2016) (internal citations and quotations omitted). As advanced below, the claims are anticipated and obvious over the prior art, especially based on the Federal Circuit's broad reasons to combine references under *KSR*.

A. SNQP1: Shteyn anticipates claims 1 and 4

1. Claim 1

(a) ***1[a] - A media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number, comprising***

Shteyn teaches a media player for receiving an audio or video program, the program comprising media data elements, from a media source over the Internet, and playing the program for a user, wherein each of the media data elements is associated with a serial number.

Shteyn teaches a computer (or other playing device) with media player software, including Java, for receiving and playing streaming media from a media source. Shteyn (Ex. 4), at 4:32-44

("A second example is to implement the method of the invention as a software application on a multi-purpose computing device, e.g., a PC or a set top box. The device has the software implementing the functionalities mentioned above."); *id.* at 3:3-7 ("In step 106, the first file segment is downloaded for play-out. Communicating with a remote server is a well known technology. For example, Java 2.0 provides a set of standard classes that enable retrieving a remote file into a buffer or as a stream."); *id.* at 3:57-61 ("When the client has selected the proper file, either the one of which the first part is represented here as in the preferred format or the one in the alternative format, the content of the first part is downloaded from the location specified and playing out is started automatically under application control."); *id.* at 3:14-21 ("In step 110, the next file segment is downloaded at the client and stored in a buffer while the previous file segment, here the first file segment, is being played out. One option is to have the downloaded files buffered in a sequence or linked list of buffers. This functionality is typically provided by the operating system of the client. For example, MS Windows family of products creates a memory buffer associated with the file every time an API call opens the file.")

Shteyn teaches that the server partitions a file into a sequence of media segments and creates a control file listing those segments. *Id.* at Abstract ("An electronic file ..., e.g., an MP3 file, is partitioned into a sequence of segments at the server side...."); *id.* at 1:65-2:7 ("To this end the content file is split into multiple parts.... The client device/application receives control information about the content. This control information comprises … information relating to the size and memory location of the whole file as well as of it[s] parts at the server."), claim 9.

Shteyn further teaches that the segments in the control file are associated with a serial number (filename or URL). *Id.* at 2:60-3:2 ("The control information describes the locations, e.g., URL's, and size of the various file segments . . . . Thus, the client is enabled to get information

about the content information and the URLs of the first and subsequent file segments.") Figure 2 below shows a control file with serial numbers (part1, part1 alt, etc.):

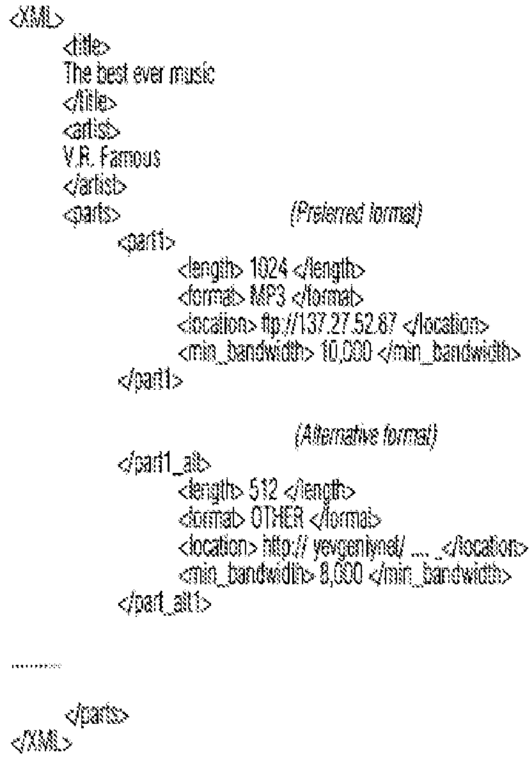


FIG. 2

Id. at Fig. 2; *id.* at 1:65-66 ("To this end the content file is split into multiple parts."), 2:25-26.

The media player then uses the control file to sequentially request each segment by serial number (URL or filename such as part 1) over the Internet. *Id.* at 4:28-31 ("[T]he client pulls the content segments from the locations indicated in the XML control information for buffering and subsequent play-out."); *id.* at 3:47-56 ("The segment labeled 'part1' is in a preferred format and described . . . the minimum bandwidth required for a connection, and the location on the Internet. An alternative first part is labeled 'part1 alt' having a different length, different format, different minimum bandwidth requirement, and a different location. . . . The client thus can automatically

choose the format compatible with the client's play-out capabilities."); *id.* at 2:57-59 ("the client contacts the server selects the particular content file and downloads the control information that enables the retrieving and playing out of the segmented file."); *id.* at 2:67-3:2 ("Thus, the client is enabled to get information about the content information and the URLs of the first and subsequent file segments."), claim 15; *id.* at 3:57-61 ("When the client has selected the proper file, either the one of which the first part is represented here as in the preferred format or the one in the alternative format, the content of the first part is downloaded from the location specified and playing out is started automatically under application control.") Figure 1 below outlines the client sequentially requesting segments by serial number while buffering and playing those segments (102-110):

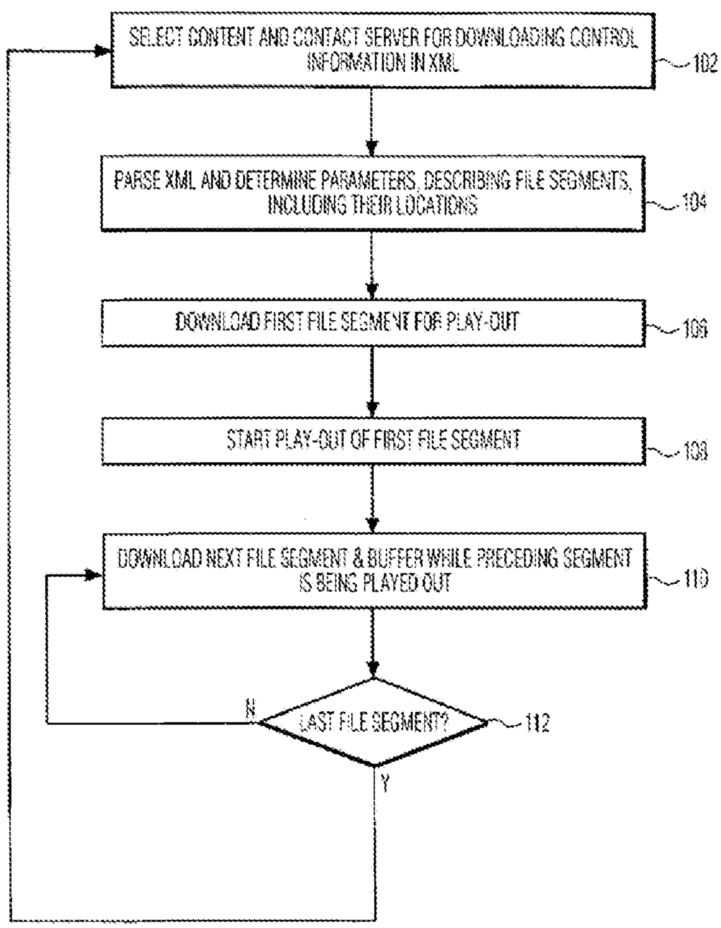


FIG. 1

Id. at Fig. 1; *id.* at 2:57-59 ("In step 102, the client contacts the server selects the particular content file and downloads the control information that enables the retrieving and playing out of the segmented file."); *id.* at 2:64-3:16 ("In step 104 the XML code is parsed. . . . Thus, the client is enabled to get information about the content information and the URLs of the first and subsequent file segments. In step 106, the first file segment is downloaded for play-out. . . . In step 108, the rendering of the first segment is started. The buffered content of the first segment is forwarded to a decoding/playing module. . . . In step 110, the next file segment is downloaded at the client and stored in a buffer while the previous file segment, here the first file segment, is being played out."), 3:63-67.

Accordingly, Shteyn teaches claim 1[a]. Further, Shteyn especially teaches claim 1[a] if WAG's proposed plain and ordinary meaning construction of "serial number / identifier" is adopted. As discussed previously, WAG contends that "serial number/identifier" is not limited to "consecutive [numbers] or limited to numerals," and thus can broadly encompass several forms. *See supra* 18. The name or label of the segment requested in Shteyn is not limited to any particular form and can even include a number in consecutive order (*e.g.*, part 1, part 2). *See* claim 1[b]-[c]; *see also* Shteyn (Ex. 4), at 1:65-66 ("To this end the content file is split into multiple parts."), 2:25-26 ("The parts can have different data formats."), 3:44-56, Fig. 2 (describing parts), claim 9. Thus, Shteyn especially anticipates under WAG's broad plain and ordinary meaning construction.

(b) ***1[b] - a processor; a memory; a connection to the network; and***

Shteyn teaches a processor, memory, and a connection to the network. Shteyn also teaches various client devices (personal computer) with hardware, including a processor. Shteyn (Ex. 4), at 4:38-42 ("A second example is to implement the method of the invention as a software application on a multi-purpose computing device, e.g., a PC or a set top box. The device has the

software implementing the functionalities mentioned above."); *id.* at 4:32-35 ("The implementation of a client in this client-server architecture can be done in a variety of ways. A first example is a hardware-based single-purpose device, similar to the Rio MP3 player by the Diamond Corp.")

Shteyn further teaches memory to buffer in sequence one media segment after another. *Id.* at 4:28-31 ("[T]he client pulls the content segments from the locations indicated in the XML control information for buffering and subsequent play-out."); *id.* at Abstract ("While the first segment is being played out, the second is being downloaded and buffered so that it is available when the play out of the first segment is completed. While playing out a current one of the segments, next one(s) of the segments are being downloaded and buffered."); *id.* at 3:8-18 ("In step 108, the rendering of the first segment is started. The buffered content of the first segment is forwarded to a decoding/playing module. . . . In step 110, the next file segment is downloaded at the client and stored in a buffer while the previous file segment, here the first file segment, is being played out. One option is to have the downloaded files buffered in a sequence or linked list of buffers.")

Shteyn also teaches a connection to the network or downloading the segments over the Internet. Shteyn describes that the sequential segments by serial number (*e.g.*, part1) are downloaded over the Internet or other network. *Id.* at 3:47-50 ("The segment labeled 'part1' is in a preferred format and described the length of the part, e.g., in bytes, the format, the minimum bandwidth required for a connection, and the location on the Internet."); *id.* at 4:20-23 ("During operation, the client application could select a next segment in a different format for the same content to adapt to changing circumstances, e.g., lower bandwidth due to network congestion."); *id.* at claim 12 ("A client device for forming a media presentation from multiple related files stored

on server computers within a computer network, comprising: means for downloading files to the client device . . . ."), claim 15; *id.* at 2:60-3:2 ("The control information describes the locations, e.g., URL's. . . . Thus, the client is enabled to get information about the content information and the URLs of the first and subsequent file segments.")

Accordingly, Shteyn teaches claim 1[b].

(c) ***1[c] - media player software comprising instructions to cause the media player to request from the media source a predetermined number of media data elements;***

Shteyn teaches that the media player software causes the media player to request from the media source a predetermined number of media data elements in multiple ways.

First, Shteyn teaches that the client requests from the server a predetermined number of media data elements by requesting one segment at a time from the control file and repeating such requests for one segment. *See* claim 1[a]; Shteyn (Ex. 4), at Abstract ("While the first segment is being played out, the second is being downloaded and buffered so that it is available when the play out of the first segment is completed. While playing out a current one of the segments, next one(s) of the segments are being downloaded and buffered."); *id.* at 3:3-16 ("In step 106, the first file segment is downloaded for play-out. . . . The buffered content of the first segment is forwarded to a decoding/playing module. . . . In step 110, the next file segment is downloaded at the client and stored in a buffer while the previous file segment, here the first file segment, is being played out."); claim 1 ("1. A method of, at a client device, forming a media presentation from multiple related files, including a control information file, stored on one or more server computers within a computer network, the method comprising acts of: downloading the control information file to the client device; . . . concurrent with the media presentation, retrieving a next file . . . .") This

"partitioning and sequential play out enables . . . to minimize latency while downloading an electronic file." *Id.* at Abstract. This is also shown in Figure 1 further below.

Second, and separately, Shteyn requests a predetermined number of media data elements by virtue of requesting segments from the control file for storing in a linked list of buffers. *Id.* at 3:14-30 ("One option is to have the **downloaded files buffered in a sequence or linked list of buffers**. This functionality is typically provided by the operating system of the client. For example, MS Windows family of products creates a memory buffer associated with the file every time an API call opens the file.") (emphasis added); *id.* at 3:31-36 ("Upon completion of playing out the first segment, the second segment is passed on from the buffer to the decoding/playing module. This can be implemented by means of, e.g., a linked list. As known, a linked list is a data structure wherein each element (here: segment) has content data and a pointer to a next element (here: next segment).").

Accordingly based on two separate grounds, Shteyn teaches claim 1[c]. Further, Shteyn especially teaches claim 1[c] if WAG's construction of "predetermined" discussed with reference to claim 1[g] as any specified data is applied here. *See supra* 18-19. Shteyn describes requesting one segment at a time and separately requesting segments from the control file for storing in the linked list of buffers. *See* above. Thus, Shteyn especially anticipates under WAG's broad plain and ordinary meaning construction given specified data is requested.

(d) ***1[d] - instructions to cause the media player to receive media data elements sent to the media player by the media source and store the media data elements in the memory;***

Shteyn teaches the media player to receive media data elements sent to the media player by the media source and store the media data elements in the memory.

As discussed in claim 1[a]-[b], the client receives the segments and stores them one by one in memory. *See also* Shteyn (Ex. 4), at Abstract ("The first segment is played out upon downloading. While the first segment is being played out, the second is being downloaded and buffered so that it is available when the play out of the first segment is completed."); *id.* at 3:8-16 ("In step 108, the rendering of the first segment is started. The buffered content of the first segment is forwarded to a decoding/playing module. . . . In step 110, the next file segment is downloaded at the client and stored in a buffer while the previous file segment, here the first file segment, is being played out.") This is shown in Figure 1 further above.

This downloading of one segment and buffering of one segment is performed in Shteyn to ensure sufficient data in the client buffer for negligible latency, which is in fact an object of Shteyn's invention. *Id.* at 1:62-66 ("It is an object of the invention to provide an open architecture solution for content delivery in a download approach that allows for a low or negligible play-out latency. To this end the content file is split into multiple parts. Each part or segment requires a relatively short download time."), Abstract.

Shteyn separately teaches that media player can implement this storing or buffering using a linked list of buffers. *Id.* at 3:31-36 ("Upon completion of playing out the first segment, the second segment is passed on from the buffer to the decoding/playing module. This can be implemented by means of, e.g., a linked list. As known, a linked list is a data structure wherein each element (here: segment) has content data and a pointer to a next element (here: next segment)."), 3:16-18 ("One option is to have the downloaded files buffered in a sequence or linked list of buffers.").

Accordingly, Shteyn teaches claim 1[d].

(e) ***1[e] - instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer;***

Shteyn teaches a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer.

Shteyn teaches a computer with a media player buffer manager, Java or operating system (Microsoft Windows), that manages the storing of segments into buffer(s). Shteyn (Ex. 4), at 3:3-7 ("In step 106, the first file segment is downloaded for play-out. Communicating with a remote server is a well known technology. For example, **Java 2.0 provides a set of standard classes that enable retrieving a remote file into a buffer or as a stream.**") (emphasis added); *id.* at 3:14-27 ("In step 110, the next file segment is downloaded at the client and stored in a buffer while the previous file segment, here the first file segment, is being played out. One option is to have the downloaded **files buffered in a sequence or linked list of buffers**. This **functionality is typically provided by the operating system** of the client. For example, **MS Windows family of products creates a memory buffer associated with the file every time an API call opens the file**. Alternatively, in a thread- and/or process-rich environment, several threads and/or processes can be organized to independently retrieve file segments, while playing out the content of other segments. . . . For example, Java 2.0 from Sun Microsystems provides classes supporting multiple threads.") (emphasis added); *id.* at 3:63-4:3 ("For example, Java JDK v.1.2 from Sun Microsystems, Inc. provides a class java.io.SequenceInputStream as a standard component of the io class library. SequenceInputStream represents the logical concatenation of other input streams. It starts out with an ordered collection of input streams and reads from the first one until end of

file is reached, whereupon it reads from the second one, and so on, until end of file is reached on the last of the contained input streams.”).

Shteyn’s buffer manager (Java or operating system such as Microsoft Windows) maintains a record of the serial number of the last element received and stored in the buffer in two ways.

First, Shteyn will record (or buffer) and play the first segment, *e.g.*, part1, while the second, *e.g.*, part2, and subsequent segments are downloaded, buffered, and played until the end of the video. *Id.* at Abstract (“The first segment is played out upon downloading. While the first segment is being played out, the second is being downloaded and buffered so that it is available when the play out of the first segment is completed. While playing out a current one of the segments, next one(s) of the segments are being downloaded and buffered.”); *id.* at 3:3-16 (“In step 106, the first file segment is downloaded for play-out. Communicating with a remote server is a well known technology. For example, Java 2.0 provides a set of standard classes that enable retrieving a remote file into a buffer or as a stream. . . . In step 110, the next file segment is downloaded at the client and stored in a buffer while the previous file segment, here the first file segment, is being played out.”); *id.* at 3:47-56 (“The segment labeled ‘part1’ . . . described . . . the minimum bandwidth required for a connection, and the location on the Internet. An alternative first part is labeled ‘part1 alt’ having a different length, different format, different minimum bandwidth requirement, and a different location. . . . The client thus can automatically choose the format compatible with the client's play-out capabilities.”); *id.* at Fig. 2 (control file with part1 and part1 alt); *id.* at claim 1; *id.* at 4:28-31 (“the client pulls the content segments from the locations indicated in the XML control information for buffering and subsequent play-out.”) Thus, Shteyn teaches that the buffer manager maintains a record of each segment by identifier (*e.g.*, part1, part2, etc.), including the last segment received, in sequence until the end of the program.

Second, and separately, Shteyn's buffer manager (operating system such as Microsoft Windows) can maintain a linked list of buffers. This data structure records in sequence each segment received by identifier (*e.g.*, part1, etc.) with a pointer to the next segment, including the last one received. *Id.* at 3:14-21 ("In step 110, the next file segment is downloaded at the client and stored in a buffer while the previous file segment, here the first file segment, is being played out. One option is to have the downloaded files buffered in a sequence or linked list of buffers. This functionality is typically provided by the operating system of the client. For example, MS Windows family of products creates a memory buffer associated with the file every time an API call opens the file."); *Id.* at 3:31-36 ("Upon completion of playing out the first segment, the second segment is passed on from the buffer to the decoding/playing module. This can be implemented by means of, e.g., a linked list. As known, a linked list is a data structure **wherein each element (here: segment) has content data and a pointer to a next element (here: next segment)**.") (emphasis added). Thus, the last downloaded element in the linked list will maintain a record of the last segment received. Accordingly, Shteyn teaches claim 1[e].

(f) ***1[f] - instructions to cause the media player to play media data elements sequentially from the player buffer; and***

Shteyn teaches that the media player plays media data elements sequentially from the player buffer. Shteyn teaches that each segment is downloaded in sequence and played out in sequence. Shteyn (Ex. 4), at Abstract ("An electronic file, e.g., an MP3 file, is partitioned into a sequence of segments at the server side. . . . While the first segment is being played out, the second is being downloaded and buffered so that it is available when the play out of the first segment is completed. . . . This partitioning and sequential play out enables . . . to minimize latency while downloading an electronic file."); *id.* at 3:14-18 ("In step 110, the next file segment is downloaded

at the client and stored in a buffer while the previous file segment, here the first file segment, is being played out. One option is to have the downloaded files buffered in a sequence or linked list of buffers."); *id.* at 3:59-65 ("the content of the first part is downloaded from the location specified and playing out is started automatically under application control. Combining multiple sequenced inputs is well understood in the industry. For example, Java JDK v.1.2 from Sun Microsystems, Inc. provides a class java.io.SequenceInputStream as a standard component of the io class library."); *id.* at 3:3-16. Figure 1 below outlines this sequential downloading and playing:

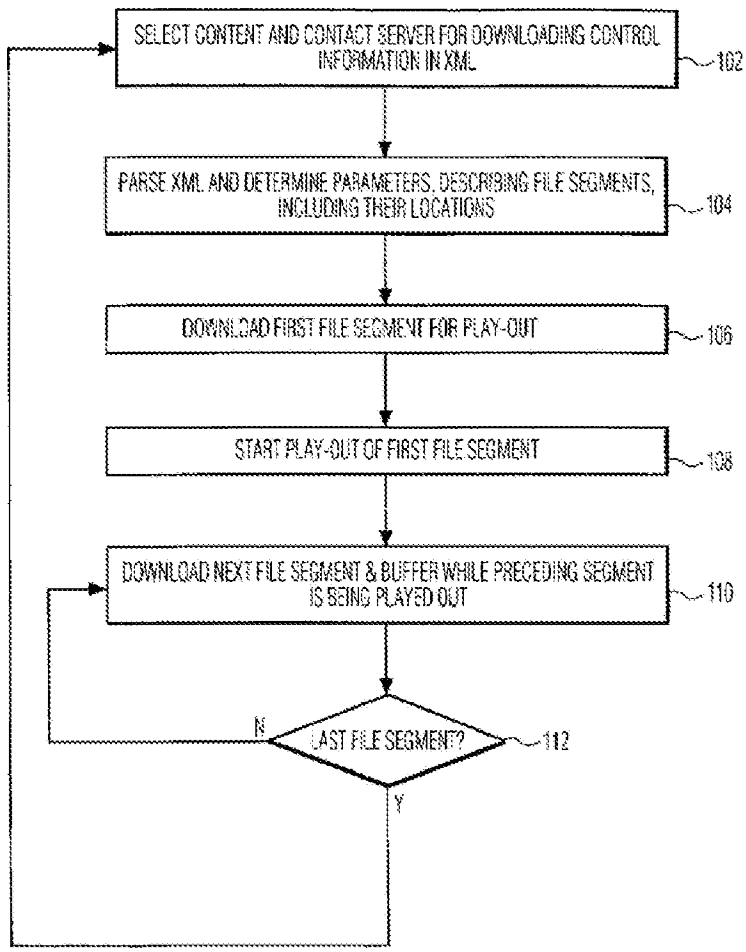


FIG. 1

Id. at Fig. 1; *id.* at 4:28-31 ("the client pulls the content segments from the locations indicated in

the XML control information for buffering and subsequent play-out.")

Accordingly, Shteyn teaches claim 1[f].

(g) ***1[g] - instructions to cause the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.***

Shteyn teaches a media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.

As discussed, Shteyn's media player will transmit to the media source a request to send one or more media data elements, each identified by a serial number. *See* claim 1[a], [c] and [e]; *see also* Shteyn (Ex. 4), at Abstract, 2:57-59, 3:3-16, 3:57-61, 3:44-56, 3:57-4:14, 4:28-31, claims 1, 15, Figs. 1-2.

Shteyn further teaches repeating these requests to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received in two ways.

First, as described in claim 1[c], Shteyn will request a predetermined one segment and buffer that one segment to minimize latency and maintain that predetermined requesting and buffering of one segment. *Id.* at Abstract ("The first segment is played out upon downloading. While the first segment is being played out, the second is being downloaded and buffered so that it is available when the play out of the first segment is completed. While playing out a current one

of the segments, next one(s) of the segments are being downloaded and buffered. This partitioning and sequential play out enables to emulate streaming of a file and to minimize latency while downloading an electronic file."); *id.* at 3:3-16 ("In step 106, the first file segment is downloaded for play-out. Communicating with a remote server is a well known technology. For example, Java 2.0 provides a set of standard classes that enable retrieving a remote file into a buffer or as a stream. In step 108, the rendering of the first segment is started. The buffered content of the first segment is forwarded to a decoding/playing module. The decoding/playing module decodes the file format, e.g., MP3. The playing of the supplied stream of bits involves a number of standard operating system calls to its drivers a technique well known in the art. In step 110, the next file segment is downloaded at the client and stored in a buffer while the previous file segment, here the first file segment, is being played out.")

Second, and separately, Shteyn maintains a predetermined number of media data elements by virtue of teaching the buffer manager (operating system such as Microsoft Windows) can maintain a linked list of buffers to store the segments repeatedly requested from the control file. *Id.* at 3:14-21 ("In step 110, the next file segment is downloaded at the client and stored in a buffer while the previous file segment, here the first file segment, is being played out. One option is to have the downloaded files buffered in a sequence or linked list of buffers. This functionality is typically provided by the operating system of the client. For example, MS Windows family of products creates a memory buffer associated with the file every time an API call opens the file."); *id.* at 3:31-36 ("Upon completion of playing out the first segment, the second segment is passed on from the buffer to the decoding/playing module. This can be implemented by means of, e.g., a linked list. As known, a linked list is a data structure wherein each element (here: segment) has content data and a pointer to a next element (here: next segment)."), Fig. 1.

This requesting and buffering of predetermined segments will continue until the end of the program. *Id.* at 3:65-4:3 ("SequenceInputStream represents the logical concatenation of other input streams. It starts out with an ordered collection of input streams and reads from the first one until end of file is reached, whereupon it reads from the second one, and so on, until end of file is reached on the last of the contained input streams."); *id.* at Abstract ("While playing out a current one of the segments, next one(s) of the segments are being downloaded and buffered."). Figure 1 shows requesting predetermined segments until the end of the program, 112 "Last File Segment":

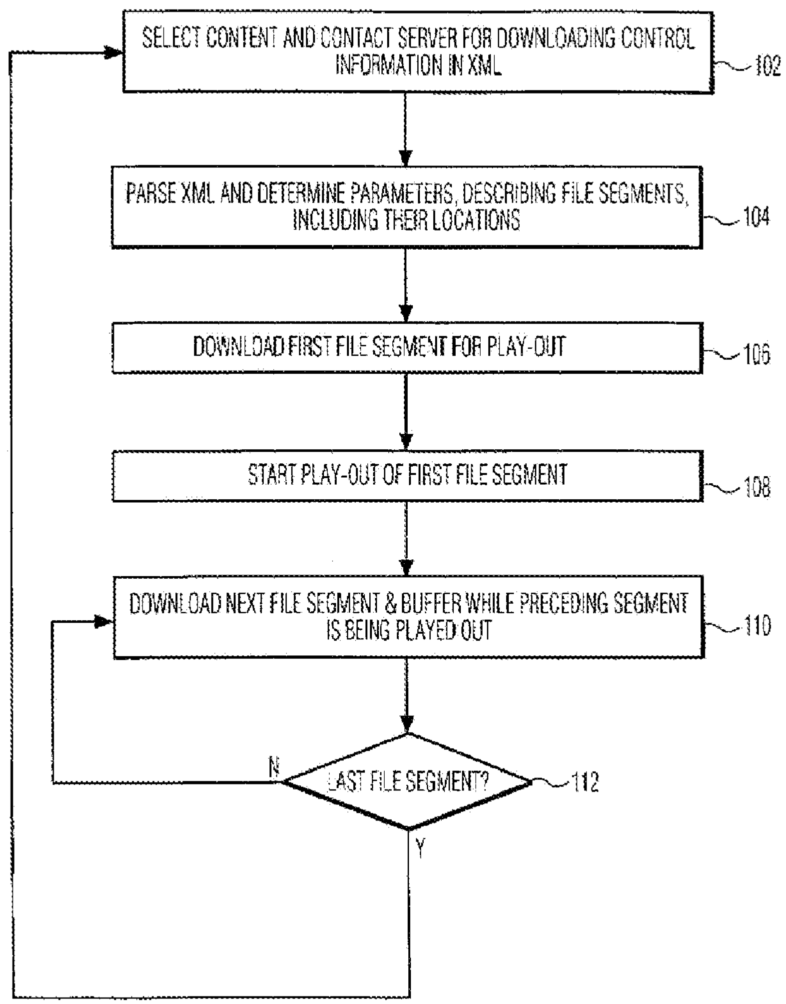


FIG. 1

Id. at Fig. 1.

Accordingly based on two separate grounds, Shteyn teaches claim 1[g]. Further, Shteyn especially teaches claim 1[g] if WAG's proposed plain and ordinary meaning construction of the claim term, "to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received" is adopted from litigation.

As discussed previously, WAG broadly contends that any specified data in the client buffer at any point (whether before the client connects to the server or after the client connects to the server) would cover this claim limitation. *See supra* 18-19 (WAG contending predetermined is a "specified number of media data elements" that "could be determined after connection to the network and after connection to the media source" and are not constant or the "same number")

Shteyn describes that it buffers a specified number of segments, *i.e.*, one segment at a time, while playing one segment until the end of the program. Shteyn (Ex. 4), Abstract ("While playing out a current one of the segments, next one(s) of the segments are being downloaded and buffered. This partitioning and sequential play out enables to emulate streaming of a file and to minimize latency while downloading an electronic file."), 3:3-30, 3:31-36, 3:63-4:3, Fig. 1. Separately, Shteyn buffers a number of segments in a linked list of buffers, while playing such segments until the end of the program. *Id.* at 3:14-30, 3:31-36, 3:63-4:3, Fig. 1. Thus, Shteyn especially anticipates under WAG's broad plain and ordinary meaning construction given the client buffers specified data.

2. **Claim 4**

Claim 4 states, "4. The media player of claim 1, wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player."

As discussed, Shteyn's media player will request and receive a predetermined number of media data elements. *See* claim 1[c] and [g]. Shteyn further teaches that the media player receives the requested predetermined segments faster than the playback rate because the client can select and receive the appropriate bandwidth segment(s) which is less than the network bandwidth.

Shteyn teaches that it provides a control file with segments of different bandwidth. Shteyn (Ex. 4), at 3:47-53 ("The segment labeled 'part1' is in a preferred format and described the length of the part, e.g., in bytes, the format, the minimum bandwidth required for a connection, and the location on the Internet. An alternative first part is labeled 'part1 alt' having a different length, different format, different minimum bandwidth requirement, and a different location.")

The client selects the appropriate bandwidth file based on network bandwidth or "system constraints" which will ensure that the network bandwidth is greater than the segment bandwidth or playback rate. *Id.* at 4:20-23 ("During operation, the client application could select a next segment in a different format for the same content to adapt to changing circumstances, e.g., lower bandwidth due to network congestion."); *id.* at claim 1 ("identifying multiple alternative flies corresponding to a given segment of the media presentation, determining which files of the multiple alternative files to retrieve based on system restraints; retrieving the determined file of the multiple alternative files to begin a media presentation . . . ."); *id.* at 2:35-38 ("The format of

parts for play-out may be chosen according to user-related information, e.g., . . . bandwidth sharing/fluctuation conditions, etc.").

Figure 2 below provides an example of segments with different bandwidth, including a part1 segment with a minimum bandwidth of 10,000 and a part1 alt segment with a minimum bandwidth of 8,000.

```
<XML>
        <title>
        The best ever music
        </title>
        <artist>
        V.R. Famous
        </artist>
        <parts>                         (Preferred format)
                <part1>
                        <length> 1024 </length>
                        <format> MP3 </format>
                        <location> ftp://137.27.52.87 </location>
                        <min_bandwidth> 10,000 </min_bandwidth>
                </part1>

                                        (Alternative format)
                </part1_alt>
                        <length> 512 </length>
                        <format> OTHER </format>
                        <location> http:// yevgenlynet/ .... </location>
                        <min_bandwidth> 8,000 </min_bandwidth>
                </part_alt1>

..........

        </parts>
</XML>
```

FIG. 2

Id. at Fig. 2. If, for instance, the network bandwidth is 9,000, then the client would select from the control file part1 alt with a bandwidth of 8,000 because the sending rate is faster than the playback rate. *Id.* at 3:55-56 ("The client thus can automatically choose the format compatible with the client's play-out capabilities."); *id.* at claim 12 (". . . . determining which file of the multiple alternative files to retrieve based on system constraints; retrieving the determined file of the

multiple alternative files to begin a media presentation . . . ."); *id.* at 3:57-61 ("When the client has selected the proper file, either the one of which the first part is represented here as in the preferred format or the one in the alternative format, the content of the first part is downloaded from the location specified and playing out is started automatically under application control.") This transmitting faster than the playback rate is also consistent with the purpose of the invention for negligible play-out latency. *Id.* at 1:62-2:3 ("It is an object of the invention to . . . for content delivery . . . that allows for a low or negligible play-out latency. . . . The size of the individual part can be determined by the communications bandwidth, e.g., through pinging for a latency-check.") If part1 was selected, it would introduce latency (video choppiness) in contrast to Shteyn's purpose because part1's bandwidth of 10,000 is above the network bandwidth of 9,000 in the example.

Indeed, the PTAB previously found a similar system (Carmel)—in which different quality or bandwidth files are selected based on network bandwidth—to Shteyn's bandwidth system to disclose sending faster than the playback rate. WebPower Final Written Decision on Remand (Ex. 12), at 21-23 ("Carmel describes client 30 making an assessment of the rate of data transfer over the link from the server and, if necessary, changing the quality level accordingly. . . . Petitioner reasons that [b]y the client selecting a lower quality level for the slices, each slice can be individually transmitted faster . . . . We accordingly find that Petitioner sufficiently identifies disclosure in Carmel that meets the last limitation of claim 10 under Carmel's lag-recovery mechanism.") (alteration in original, citations and quotations omitted). Thus, Shteyn also discloses the media player receiving data faster than the playback rate like Carmel.

Accordingly, Shteyn teaches claim 4.

B. SNQP2:Shteyn and Carmel, Glaser, or TCP render obvious claims 1 and 4

1. Motivation to Combine

A POSITA would have found it obvious to combine the teachings of Carmel and Shteyn because the combination is merely a matter of applying a known technique to a known device for improvement to yield predictable results, including for claim 4.

Shteyn and Carmel disclose similar client-pull systems, including requesting data by serial number using a control or index file. *See supra* 11-13; WebPower Final Written Decision on Remand (Ex. 12), at 16-23. They also both teach the client's ability to select different quality or bandwidth files from the index file depending on network bandwidth. *Compare* Shteyn (Ex. 4), at 3:44-53, 3:55-56, 4:20-26, Fig. 2, claims 1, 12, *with Carmel* (Ex. 3), at 2:24-27, 3:5-9, 7:24-27, 8:42-9:5, Fig. 3D.

A POSITA under *KSR* (*see supra* 20-21) at the time of the purported invention was made would have been motivated to combine the teachings of Carmel and Shteyn given the similarity of the Shteyn and Carmel systems and a POSITA would have been motivated to review other client-pull systems to further efficiency and enhance Shteyn's functionality. This includes, for instance, implementing Carmel's sending faster than playback rate in Shteyn.

A POSITA would have been further motivated to implement such function and would have had a reasonable expectation of success in doing so considering Shteyn's existing server and client functionality. Shteyn has client buffers (or linked list of buffers) such that the predetermined segments could be received faster than the playback rate. Shteyn (Ex. 4), at Abstract, 1:6-14, 3:31-36, 3:14-18. Moreover, Shteyn has an existing system, just like Carmel, to select the appropriate segment depending on network bandwidth. *Id.* at 3:44-53, 3:55-56, 4:20-26, Fig. 2, claims 1, 12. Further, Shteyn teaches that the server already partitions media into segments and that the

invention could be used for real-time communications such as broadcasts to a plurality of users. Thus, Shteyn could implement Carmel's sending faster than playback rate seamlessly.

Also, a POSITA would have been motivated to combine the references given the patentee's admissions that the TCP protocol was well-known for proper data delivery from server to client and that media players, including client buffers, were well-known for playback. *See supra* 2-3; *Unwired Planet*, 841 F.3d at 1003 ("[T]he Court [in *KSR*] explained that a rationale to combine could arise from interrelated teachings of . . . the background knowledge possessed by a person having ordinary skill in the art."); *McCoy v. Heal Sys., LLC*, 850 F. App'x 785, 789 (Fed. Cir. 2021) ("By characterizing certain parts as conventional in the specification, the patentee effectively admits that such things would be known to a POSA.") (unpublished). A POSITA would have been motivated to improve on this known knowledge, *e.g.*, transmitting data and storing such data in the client's buffer, by combining Carmel's transmitting faster than the playback rate to further enhance Shteyn's efficiency in buffering and playing streaming data. *Constant v. Advanced Micro Devices, Inc.*, 848 F.2d 1560, 1570 (Fed. Cir. 1988) ("A statement in a patent that something is in the prior art is binding on the applicant and patentee for determinations of anticipation and obviousness.") (internal citation omitted); *WesternGeco LLC v. ION Geophysical Corp. (In re WesternGeco LLC)*, 889 F.3d 1308, 1329 (Fed. Cir. 2018) ("The Board's consideration of the '967 Patent's characterization of the prior art was also proper.") (collecting cases). Accordingly, a POSITA would have been motivated to combine Shteyn and Carmel.

Additionally, a POSITA would have been motivated to combine Glaser and Shteyn for claim 4. A POSITA would have been motivated to implement Glaser's sending faster than the playback rate in Shteyn to enhance Shteyn's functionality, including real-time playback. *Id.* at Abstract, 1:62-66. Also, Glaser is cited on the front page of the '011 Patent, so a POSITA would

have been motivated to review and apply it to other prior art, including Shteyn. A POSITA would have been further motivated to implement such function considering Shteyn's existing server structure and buffering ability to store the data sent faster than the playback rate. *Id.* at Abstract, 1:62-66, 3:14-18, 3:31-36, 4:20-31. Moreover, Shteyn has an existing system, just like Glaser, to receive the appropriate quality data. *Compare* Shteyn (Ex. 4), at 3:44-53, 3:57-61, 4:20-26, Fig. 1, *with* Glaser (Ex. 11), at 21:57-63, 22:41-59. Accordingly, a POSITA would have been motivated to combine Shteyn and Glaser.

2. **Claim 4**

Shteyn and Carmel teach the media player receives the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player. As discussed above, a POSITA would have been motivated to combine Shteyn and Carmel for claim 4.

Shteyn already describes the client receiving a predetermined number of segments. *See supra* 34-37.

The PTAB previously found Carmel anticipated claim 10 of the '141 patent, including sending the requested data at a rate more rapid than the rate at which the data is played back. WebPower Final Written Decision on Remand (Ex. 12), at 16-23 ("The parties dispute only . . . whether Carmel discloses . . . to cause the server to send media data elements to the user system responsive to said requests, at a rate more rapid than the rate at which said streaming media is played back by a user, in light of the Federal Circuit's construction of rate. . . . Because we find that Petitioner identifies sufficient disclosure in Carmel to meet that limitation, under either normal operation or with a lag-recovery mechanism, we conclude that Petitioner shows, by a preponderance of the evidence, that independent claim 10 is anticipated by Carmel.") (citations

and quotations omitted); '141 patent (Ex. 1), claim 10 ("routine containing instructions to cause the server to send media data elements to the user system responsive to said requests, at a rate more rapid than the rate at which said streaming media is played back by a user."). Because WAG never appealed this PTAB determination, it is an admitted fact that Carmel teaches this limitation.

The PTAB found—under the Federal Circuit's construction of "rate"—that Carmel transmits slices faster than playback rate to the client in two different ways. First, the PTAB determined that in Carmel's normal operation it expressly discusses sending slices at a rate more rapid than the playback rate:

> To support its contention that Carmel teaches sending media data elements at a rate more rapid than the playback rate during "normal operation," Petitioner quotes the following disclosure: In some preferred embodiments of the present invention, the transmitting computer and the clients monitor the uploading and downloading of data to and from the server, respectively, in order to determine **the amount of time required to convey each slice and to verify that the slices are conveyed at a sufficient rate.** When the data stream comprises multimedia data, **the data rate should be generally equal to or faster** than the rate at which the data are generated at the transmitting computer. Pet. Remand Br. 3 (quoting Ex. 1003, 2:51–59) (emphasis by Petitioner). As Petitioner correctly observes, "this passage is describing the 'sufficient rate' during normal streaming operation (uploading and downloading of data to and from the server), not the later-described embodiments using multiple links as one way to compensate for lag or slow connections." *Id.* at 3–4. Petitioner reasons that the last limitation of independent claim 10 is met because Carmel explicitly teaches that such normal operation may use a data rate that is "faster" than the rate at which the data are generated by the transmitting computer. *Id.*
>
> In addition to this intrinsic evidence, Petitioner further supports its contention by pointing to testimony by Patent Owner's expert, Mung Chiang, Ph.D. See *id.* at 5–6. On cross examination, Dr. Chiang explained that "Carmel adjusts the slices so that they are transmitted at about the playback rate." Ex. 1022, 91:10–12. When Petitioner explored the consequences of what Dr. Chiang meant in describing transmission of slices "at about the playback rate," <u>Dr. Chiang conceded that "[i]f it is transmitted slightly faster than playback rate</u> and then slightly lower, slightly higher, slightly lower, which is what 'about playback rate' means." *Id.* at 92:16–19.

> Petitioner argues that, because Carmel teaches that transmission occurs, at least sometimes, at a rate greater than the playback rate, the last limitation of claim 10 is met. Pet. Remand Br. 6 ("This is a dispositive admission—transmitting at least sometimes 'slightly faster' than playback meets the claim limitation of sending 'one or more' requested data elements faster than playback is met."). In advancing that argument, Petitioner relies on *Broadcom Corporation v. Emulex Corporation*, 732 F.3d 1325, 1333 (Fed. Cir. 2013) for the proposition that a claim limitation is met by a device that performs the function "some of the time." *Id.*
>
> . . . .
>
> [Carmel] contains instructions for sending media data elements at a rate more rapid than the playback rate. We accordingly find that Petitioner sufficiently identifies disclosure in Carmel that meets the last limitation of claim 10 under Carmel's normal operation.

WebPower Final Written Decision on Remand (Ex. 12), at 18-21 (bold emphasis in original by Petitioner; underline emphasis added); Mung Chiang, Ph.D., *WebPower v. WAG Acquisition*, IPR2016-01238, Ex. 1022, at 92:16–19 (Jun. 20, 2017) ("If it is transmitted slightly faster than playback rate and then slightly lower, slightly higher, slightly lower, which is what 'about playback rate' means . . . ."); *id.* at 91:10–12; Carmel (Ex. 3), at 2:51-59 ("[T]he transmitting computer and the clients monitor the uploading and downloading of data to and from the server, respectively, in order to determine the amount of time required to **convey each slice** and to verify that the slices are conveyed at a sufficient rate. When the data stream comprises multimedia data, the **data rate should be generally** equal to or **faster than the rate at which the data are generated at the transmitting computer**.") (emphasis added).

Second, and separately, the PTAB found that Carmel teaches the rate more rapid than the playback rate limitation under its lag-recovery mechanism because if a lag is detected, Carmel selects lower quality level slices thereby transmitting each slice faster:

> As an alternative, Petitioner contends that "Carmel further discloses a faster-than-playback transmission rate when it states that '[i]n the event that a lag is detected, steps are taken to **increase the data transmission or reception rate.'"** Pet. Remand Br. 8 (quoting Ex. 1003, 7:39–42)

> (alterations by Petitioner). In particular, Carmel describes client 30 making an assessment of the rate of data transfer over the link from the server and, if necessary, changing the quality level accordingly. Ex. 1003, 11:9–11. "For example, if the rate is low, such that the time stamps 59 indicate that the slices need to be played as fast or faster than they are being received, the client will preferably select a lower quality level." Id. at 11:11–15.
>
> Petitioner reasons that "[b]y the client selecting a lower quality level for the slices, each slice can be individually transmitted faster, and the slices would no longer be 'played as fast or faster than they are being received' thus recovering from lag." Pet. Remand Br. 9. After such a correction, the transmission of the media data elements is faster than the media being played, thereby meeting the claim limitation. *Id.*
>
> . . . .
>
> Carmel explains that "the level #1 slices have smaller data volume than the level #2 slices and can therefore be transmitted over a lower-bandwidth data link, while maintaining the required timing indicated by time stamps 59." *Id.* at 8:57–66. Thus, Petitioner reasons, "when the client determines a new quality level is required, the client will request the lower quality level for that slice ID. Pet. Remand Br. 9–10 (citing Ex. 1003, 11:11–15 ("the client will preferably select a lower quality level")). We agree that Carmel's disclosure accordingly supports Petitioner's assertion that "[t]he server does not ever send new or different files than the ones requested by clients," *id.* at 10; rather each of the clients "chooses . . . the quality level appropriate to the bandwidth of its link on network 28 to server 36," Ex. 1003, 9:6–9.
>
> . . . .
>
> We accordingly find that Petitioner sufficiently identifies disclosure in Carmel that meets the last limitation of claim 10 under Carmel's lag-recovery mechanism.

WebPower Final Written Decision on Remand (Ex. 12), at 21-23 (bold emphasis in original; underline emphasis added); *see also* Carmel (Ex. 3), at 7:40-41 ("In the event that a lag is detected, steps are taken to increase the data transmission or reception rate . . . ."), 11:9-15 ("Periodically, client 30 makes an assessment of the rate of data transfer over the link from the server and, if necessary, changes the quality level accordingly. For example, if **the rate is low, such that time stamps 59 indicate that the slices need to be played as fast as or faster than they are being received, the client will preferably select a lower quality level** if one is available.") (emphasis added). Accordingly, Shteyn and Carmel teach claim 4.

Also, a POSITA would have been motivated to combine TCP with Shteyn given WAG's admission that TCP "sends" or transmits data faster than the playback rate. '011 Patent (Ex. 2), at 6:10-16 (media data is "**sent as fast as the communications medium will support**, **either by the server buffer passing media data to the transport mechanism**, or by **the transport mechanism delivering or redelivering the media data to the user computer**.") (emphasis added), 6:10-21 (TCP sends), 4:61-67 ("This invention presumes the existence of a data communications transport mechanism, such as the TCP protocol....Thus, the delivery of data in the proper sequence is outside the scope of this invention."); *see supra* 3, 42. Thus, Shteyn and TCP teach claim 1[d] and 24[c].

Lastly, Shteyn and Glaser teach claim 4. Shteyn already describes the client receiving a predetermined number of segments. *See supra* 34-37. Glaser teaches the client buffer receives data at a rate faster than the playback rate to maintain a predetermined threshold. Glaser (Ex. 9), at 22:57-59 ("As the normal quality data blocks are transmitted at greater than real time, the buffer 315 begins to refill and approach maximum capacity."); *id.* at 21:57-60 ("When the subscriber PC 110 determines that the buffers 315 are near maximum capacity (e.g., above 85% of capacity), this indicates that the normal quality data is being transferred in real time or greater than real time."); *id.* at 20:8-16 ("audio-on-demand system 100 allows for greater than real time delivery of audio data to the subscriber PC 110 in many cases. . . ."). Thus, Shteyn and Glaser teach claim 4.

C. SNQP3: Hill anticipates or renders obvious with Carmel, Glaser, or TCP claims 1 and 4

1. Motivation to Combine

A POSITA would have found it obvious to combine the teachings of Carmel and Hill because the combination is merely a matter of applying a known technique to a known device for improvement to yield predictable results, including for claim 4.

Hill and Carmel both teach video distribution systems over the Internet, including requesting data by serial number and to repeat such requests. *See supra* 11-13; WebPower Final Written Decision on Remand (Ex. 12), at 16-23. Further, both maintain a record of the last serial number received. *Id.* A POSITA under *KSR* (*see supra* 20-21) at the time of the purported invention was made would have been motivated to combine the similar systems. For instance, a POSITA would have been motivated to implement Carmel's sending faster than playback rate to further enhance Hill's playback. Hill (Ex. 10), at 6:58-67.

A POSITA would have been further motivated to implement such function and would have had a reasonable expectation of success in doing so considering Hill's existing functionality. Hill already discloses maintaining a request threshold number of frames based on system characteristics (*id.* at 1:7-11, 6:58-67, 10:55-60) and the server sending 48 frames per second or twice the rate of the playback rate at 24 frames per second. *Id.* at 7:12-24. Moreover, Hill teaches that its client buffering system adapts to data sources that vary in speed and response time. *Id.* at Abstract, 1:7-11. Thus, Hill could implement Carmel's sending faster than playback given the existing server and client structure in Hill and overlapping similarities discussed above.

Also, a POSITA would have been motivated to combine the references given the patentee's admissions that the TCP protocol was well-known for proper data delivery from server to client and that media players, including client buffers, were well-known for playback. *See supra* 2-3; *Unwired Planet*, 841 F.3d at 1003; *McCoy*, 850 F. App'x at 789. A POSITA would have been motivated to improve on this known knowledge, *e.g.*, transmitting data and storing such data in the client's buffer, by combining Carmel's transmitting faster than the playback rate to further enhance Hill's efficiency in buffering and playing streaming data. *Advanced Micro Devices*, 848

F.2d at 1570; *WesternGeco*, 889 F.3d at 1308. Accordingly, a POSITA would have been motivated to combine Hill and Carmel.

Additionally, a POSITA would have been motivated to combine Glaser with Hill. Hill and Glaser both teach video distribution systems over the Internet. *See supra* 11-12, 15. Glaser transmits multimedia data faster than the playback rate to maintain a client buffer threshold. Glaser (Ex. 9), at 3:15-28, 3:31-36, 13:36-44, 16:2-10, 21:57-63, 22:41-59. A POSITA *KSR* (*see supra* 20-21) would have been motivated to implement Glaser's sending data faster than the playback rate in Hill to enhance Hill's functionality, including playback. Hill (Ex. 10), at Abstract, 6:58-67. Also, Glaser is cited on the front page of the '011 Patent, so a POSITA would have been motivated to review and apply it to other prior art, including Hill.

A POSITA would have been further motivated to implement such function considering Hill's existing functionality. Hill already teaches maintaining a threshold of data and sending at twice the rate of the playback rate. Moreover, Hill teaches that its client buffering system adapts to data sources that vary in speed and response time. *Id.* at Abstract, 1:7-11. Thus, Hill could buffer the data sent by Glaser faster than the playback rate. Accordingly, a POSITA would have been motivated to combine Hill and Glaser.

2. **Claim 1**

(a) ***1[a] - A media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number, comprising***

Hill teaches a media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is

associated with a serial number.

Hill describes a media player receiving a clip or clips in a sequence of frames over the Internet and playing those frames. Hill (Ex. 10), at 1:7-11 ("The present invention relates generally to audio/video playback devices, and more particularly to playing movie data held in various storage devices with varying bandwidth capabilities, where the movie player adapts to these varying conditions."); *id.* at 3:46-56 ("A clip is a sequence of frames. The movie data of interest to a user will be composed of one or more clips. Clips may be stored in different locations within the digital network environment 102. For purposes of discussion, assume that the movie data of interest to the user contains Nc clips, each stored in a separate source 122. . . . Each source 122 may be any type of digital memory, and may be accessed either locally or via a network server. Note that a source may be a site on the Internet."); *id.* at 3:28-32 ("A user 106 utilizes the workstation 104 to play movie data. The workstation 104 may be a conventional personal computer, a high performance workstation, or dedicated hardware specially suited for audio/video playback."); *id.* at 3:37-39 ("Movie data is composed of synchronized audio and video information. This information is broken into fundamental units called frames.")

Hill further describes that the media player requests each frame by the frame number, where each frame is associated with a frame or serial number. *Id.* at 7:43-50 ("The buffer manager 118 does this by checking to see if it has frames 21 through 31. Normally, the most distant frame has not been requested. Assuming this is the case, the buffer manager 118 will then request global frame 31. This request is sent to the cliplist manager, which maps the global frame number to a clip number and local frame number and forwards the request to the appropriate source."); *id.* at 5:64-6:16 ("The three clips together form the global playback sequence. Frames may either be referenced according to their position within the global playback sequence (i.e., global frame

number), or according to their position within the clip from which they originate (i.e., local frame number). . . . In step 310, the buffer manager 118 sends requests for new frames to the cliplist manager 116. The cliplist manager 116 maintains the mapping between global and local frame numbers. Thus, in step 312, the cliplist manager 116 converts the global frame number to a clip number and local frame number. In step 314, the cliplist manager 116 requests the frames from the appropriate source manager 120. The source manager 120 in step 316 retrieves the requested frame data from the source 122."); *id.* at 12:7-13 ("This example cliplist has three clips. The first clip is local frames 50 through 138 from the file '/sequence1/scene2/latest.movie' . . . and the third clip is local frames 0 through 50 in '/sequence1/scene4/latest.movie' . . . . The first and third are located on server1.").

The frames associated with frame numbers are shown in Figures 6B and 7 below:

FIG. 6B

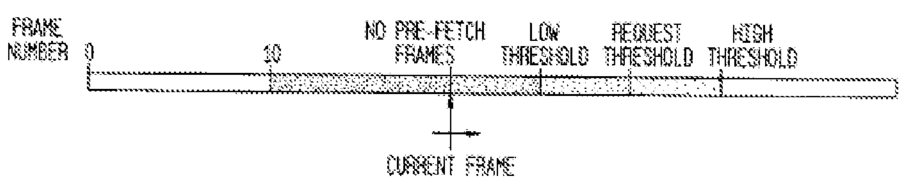
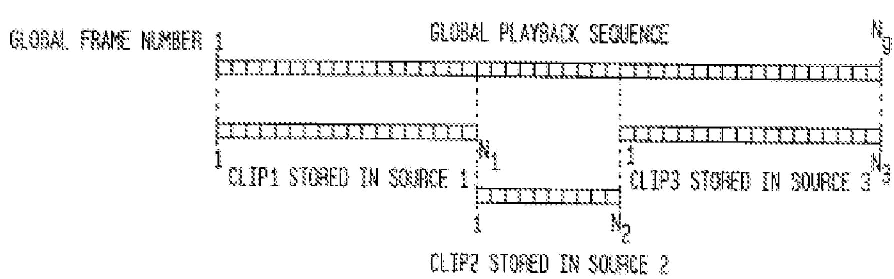


FIG. 7



Id. at Figs. 6B, 7; *id.* at 3:6-11 ("FIG. 6 summarizes the various thresholds applied to the buffer fill level according to the preferred buffer management strategy; and FIG. 7 represents the organization and representation of movie data within the preferred embodiment.")

Accordingly, Hill teaches claim 1[a]. Further, Hill especially teaches 1[a] if WAG's proposed plain and ordinary meaning construction of "serial number / identifier" is adopted. As discussed previously, WAG contends that "serial number/identifier" is not limited to "consecutive [numbers] or limited to numerals," and thus can broadly encompass several forms. *See supra* 18. Hill teaches that the identifier associated with the requested frames is not limited to any particular form and indeed, the frame numbers can be numeric and consecutive. Hill (Ex. 10), at 5:60-6:16, 7:39-50, 11:63-12:13, Figs. 6B, 7. Thus, Hill especially anticipates under WAG's broad plain and ordinary meaning construction.

(b) ***1[b] - a processor; a memory; a connection to the network; and***

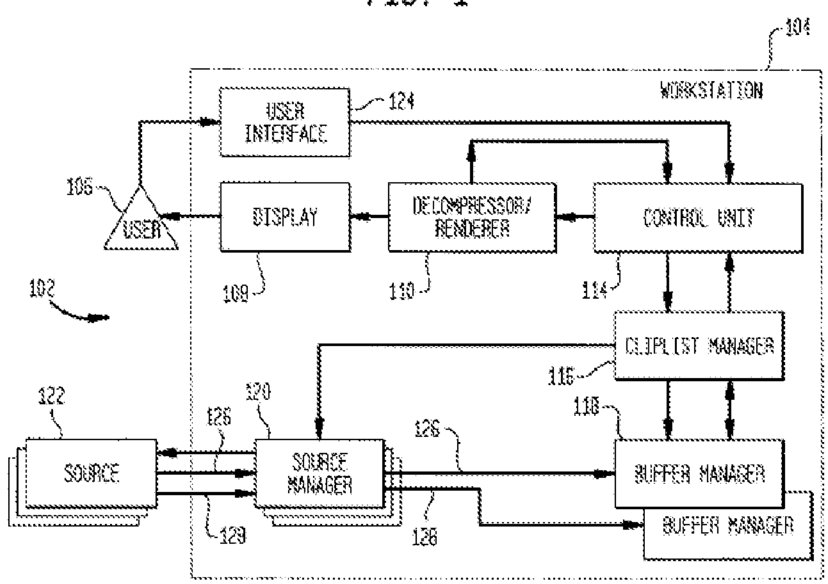
Hill teaches a processor, memory, and connection to the network.

Hill describes a user workstation that has a control unit or processor, memory and hard drive for buffering, and connection to the network such as the Internet. Hill (Ex. 10), at 3:57-64 ("The workstation 104 contains a control unit 114 responsible for the overall control of the system. The control unit 114 receives control information from the user 106 via the user interface 124. . . . The control unit 114 also receives control information from the decompressor/renderer 110. This control path will be discussed below."); *id.* at 4:7-15 ("The buffer manager 118 provides frames upon request to the control unit 114 via the cliplist manager 116. In order to provide frames quickly, the buffer manager 118 stores a limited number of frames in local digital memory, such as random access memory (RAM) or a local disk drive."); *id.* at 4:16-20 ("The buffer manager 118 also retains a certain number of the frames most recently sent to the control unit 114. This allows

the buffer manager 118 to respond immediately if the direction of play is reversed. These frames will be referred to as history frames."); *id.* at claim 1 ("A system for playing digital movie data under the control of a user, wherein the movie data comprises a sequence of frames, said system comprising: . . . buffer management means, coupled to said control means and to said source means, for storing a subset of the movie data and providing the movie data as needed to said control means, said buffer management means comprising: local digital memory that stores a plurality of frames including future frames and history frames . . . ."); *id.* at Abstract ("The present invention is directed to a system and method for enabling a user to view a moving picture with synchronized audio using a workstation connected to a digital network."); *id.* at 3:48-56 ("Clips may be stored in different locations within the digital network environment 102. For purposes of discussion, assume that the movie data of interest to the user contains Nc clips, each stored in a separate source 122. . . . Each source 122 may be any type of digital memory, and may be accessed either locally or via a network server. Note that a source may be a site on the Internet.")

The user's computer with the connection over the Internet is shown in Figure 1 below.

FIG. 1



Id. at Fig. 1.

Accordingly, Hill teaches claim 1[b].

(c) ***1[c] - media player software comprising instructions to cause the media player to request from the media source a predetermined number of media data elements;***

Hill teaches a media player to request from the media source a predetermined number of media data elements.

As discussed, Hill will request frames by frame or serial number from the source. *See* claim 1[a]; *see also Id.* at 5:63-6:16, 11:63-12:13, Figs. 6B, 7.

Hill describes that the media player's buffer manager (using the cliplist manager and source manager) will request a predetermined number of frames, either one or two frames from the source, to maintain a "request threshold." Hill (Ex. 10), at 7:12-17 ("The buffer manager 118 will request **one new frame** from the cliplist manager 116 if the request threshold exceeds the fill level by one frame (i.e., one frame behind). If the buffer manager 118 is more than one frame behind, it will **request two new frames**. . . .") (emphasis added); *id.* at 5:52-54, 9:40-44 ("the buffer manager 118 requests two new frames when the fill level is two frames less than the request threshold, one frame when the fill level is one frame less than the request threshold, and no frames requested otherwise."); *id.* at 6:7-16 ("In step 310, the buffer manager 118 sends requests for new frames to the cliplist manager 116. The cliplist manager 116 maintains the mapping between global and local frame numbers. Thus, in step 312, the cliplist manager 116 converts the global frame number to a clip number and local frame number. In step 314, the cliplist manager 116 requests the frames from the appropriate source manager 120. The source manager 120 in step 316 retrieves the requested frame data from the source 122."); *id.* at 7:46-50 ("the buffer manager 118 will then request global frame 31. This request is sent to the cliplist manager, which maps the global frame

number to a clip number and local frame number and forwards the request to the appropriate source.”). The “request threshold represents the optimum number of future frames stored in the buffer at any given time, and should be determined based on the characteristics of a particular system (e.g., I/O bandwidth, source latency).” *Id.* at 6:63-67.

Accordingly, Hill teaches claim 1[c]. Further, Hill especially teaches claim 1[c] if WAG’s construction of “predetermined” discussed with reference to claim 1[g] as any specified data is applied here. *See supra* 18-19. Hill requests one or two frames repeatedly to maintain a request threshold. *See* above. Thus, Hill especially anticipates under WAG’s broad plain and ordinary meaning construction given specified data is requested.

(d) ***1[d] - instructions to cause the media player to receive media data elements sent to the media player by the media source and store the media data elements in the memory;***

Hill teaches the media player to receive media data elements sent to the media player by the media source and store the media data elements in the memory.

As discussed in claim 1[a]-[c], the media player (buffer manager) will receive frames associated with frame numbers and buffer those frames in memory, including maintaining a history. *See also* Hill (Ex. 10), at 5:24-30 (“buffer manager 118 responds to requests for frames from cliplist manager 116 then determines whether new frames need to be loaded. If new frames are required, it uses the cliplist manager 116 to retrieve these frames from the appropriate source 122. The source 122 accesses the requested frame then requests space from the buffer manager 118 before transferring it.”); *id.* at 7:1-2 (“buffer manager 118 retains a number of history frames equal to the request threshold as well”); *id.* at 10:55-57 (“the buffer manager 118 retains the n_h frames most recently sent to the control unit 114 (i.e., history frames), where n_h equals the request threshold.”); *id.* at claim 1 (“A system for playing digital movie data under the control of

a user, wherein the movie data comprises a sequence of frames, said system comprising: . . . buffer management means, coupled to said control means and to said source means, for storing a subset of the movie data and providing the movie data as needed to said control means, said buffer management means comprising: local digital memory that stores a plurality of frames including future frames and history frames . . . ."); *id.* at 1:7-11 ("The present invention relates generally to audio/video playback devices, and more particularly to playing movie data held in various storage devices with varying bandwidth capabilities, where the movie player adapts to these varying conditions.").

Accordingly, Hill teaches claim 1[d].

(e) ***1[e] - instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer;***

Hill teaches a player buffer manager for managing a player buffer to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer.

As discussed, Hill teaches a player buffer manager that requests frames by frame or serial number and buffers such frames, including maintaining a history, in memory. *See* claim 1[a], [c], and [d]; *see also* Hill (Ex. 10), at 5:63-6:16, 6:58-67, 11:63-12:13.

Hill will maintain a record of the last frame number received because the media player (buffer manager) uses the last frame number received to request the next frame number. *Id.* at 10:4-6 ("the buffer manager 118 normally uses the **global frame number last requested as the starting point for determining which frame to request next**.") (emphasis added). Hill provides

an example of the buffer manager recording frame numbers, including the last one received, and using that last frame number to request the next frame number:

> When the control unit 114 requests global frame 20 (via the cliplist manager 116), the buffer manager 118 replies by providing global frame 20 to control unit 114 via cliplist manager 116 and **checks the fill level against the request threshold** (for this example, request threshold=10). The **buffer manager 118 does this by checking to see if it has frames 21 through 31**. Normally, the most distant frame has not been requested. Assuming this is the case, **the buffer manager 118 will then request global frame 31**. This request is sent to the cliplist manager, which maps the global frame number to a clip number and local frame number and forwards the request to the appropriate source.

Id. at 7:39-50 (emphasis added).

Additionally, and separately, Hill teaches that the buffer manager maintains a history of the frames received, including the last one received. *Id.* at 7:1-2 ("buffer manager 118 retains a number of history frames equal to the request threshold as well. . . ."); *id.* at 10:55-60 ("the buffer manager 118 retains the n_h frames most recently sent to the control unit 114 (i.e., history frames), where n_h equals the request threshold. Thus, the buffer manager 118 retains as many history frames as the optimum number of future frames. This situation is illustrated by FIG. 5(a) and FIG. 6(b)."); *id.* at 11:42-45 ("The buffer manager 118 preferably takes advantage of this opportunity to ensure that the optimum number of future and history frames are currently being stored.") This frame history is shown in Figure 5(a) and 6(b) below:

FIG. 5A

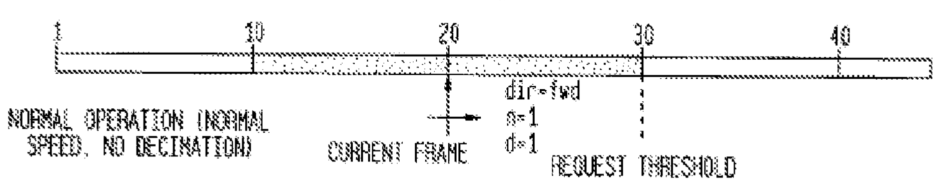
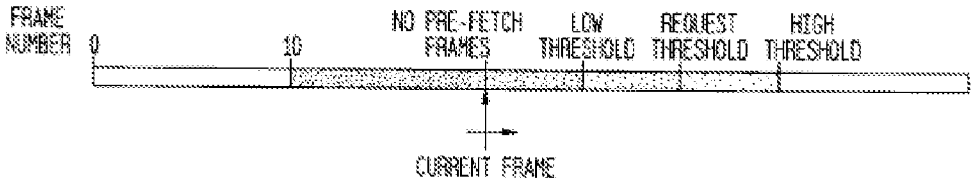


FIG. 6B



Id. at Figs. 5(a), 6.

Accordingly, Hill teaches claim 1[e].

(f) ***1[f] - instructions to cause the media player to play media data elements sequentially from the player buffer; and***

Hill teaches the media player to play media data elements sequentially from the media player buffer. *See* claim 1[a].

Specifically, Hill describes that a "clip is a sequence of frames" which are played sequentially from the player buffer. Hill (Ex. 10), at 3:46-56; *id.* at claim 12 ("A workstation for playing digital movie data . . . wherein each of the tracks comprises a sequence of synchronized frames, and wherein at least one remote source means stores a portion of the movie data, the workstation comprising: display means for displaying the movie data to said user . . . a plurality of buffer management means, one for each of said tracks, coupled to said control means and to the remote source means, for storing a subset of the movie data and providing the movie data as needed to said control means, said buffer management means comprising: local digital memory that stores a plurality of frames including future frames and history frames, prefetch means for requesting future frames from the remote source means . . . ."); *id.* at 3:28-32 ("A user 106 utilizes the workstation 104 to play movie data. The workstation 104 may be a conventional personal computer, a high performance workstation, or dedicated hardware specially suited for audio/video playback."); *id.* at 5:60-65 ("The control unit 114 and the buffer manager 118 reference frames

according to a global frame number, as shown in FIG. 7. Here, three clips are shown (Nc =3), each of a different length (N1, N2, N3) and stored in a different source. The three clips together form the global playback sequence."); *id.* at 11:63-64 ("[T]he current invention accesses a global playback sequence consisting of several clips."); *id.* at Abstract ("The present invention is directed to a system and method for enabling a user to view a moving picture with synchronized audio using a workstation connected to a digital network.")

Accordingly, Hill teaches claim 1[f].

(g) ***1[g] - instructions to cause the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.***

Hill teaches the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.

Hill's media player will transmit to the media source a request to send one or more media data elements, each identified by a serial number. *See* claim 1[a] and [c].

Hill further teaches that it repeats requests for frames (*e.g.*, one or two frames) to maintain a specified frames in the player buffer or "request threshold." Hill (Ex. 10), at 9:40-44 ("the buffer manager 118 requests two new frames when the fill level is two frames less than the request threshold, one frame when the fill level is one frame less than the request threshold, and no frames requested otherwise."); *id.* at 6:58-67 ("Referring to FIG. 4A, the buffer manager 118 determines

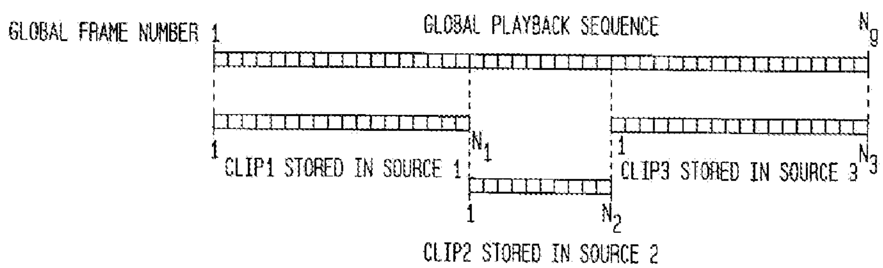
whether the buffer fill level has dropped below the request threshold, as shown in step 404. . . . The request threshold represents the optimum number of future frames stored in the buffer at any given time, and should be determined based on the characteristics of a particular system (e.g., I/O bandwidth, source latency)."); *id.* at claim 9 ("The system of claim 8, wherein said prefetch means requests no frames if the buffer fill level is greater than said request threshold, one frame if the buffer fill level is equal to or one frame less than the request threshold, and two frames if the buffer fill level is two frames or more less than the request threshold."); *id.* at claim 8 ("The system of claim 1, wherein said prefetch means compares the buffer fill level to a request threshold and requests a select number of frames based on this comparison."); *id.* at 5:52-54 ("In a preferred embodiment, the buffer manager 118 either makes no request, requests one frame, or requests two frames."); *id.* at 7:39-43 ("When the control unit 114 requests global frame 20 (via the cliplist manager 116), the buffer manager 118 replies by providing global frame 20 to control unit 114 via cliplist manager 116 and checks the fill level against the request threshold (for this example, request threshold=10).").

And this predetermined number is maintained for the duration of the program or movie (*i.e.*, "one or more clips"). *Id.* at 3:46-47 ("A clip is a sequence of frames. The movie data of interest to a user will be composed of one or more clips."); *id.* at 6:63-67 ("The request threshold represents the optimum number of future frames stored in the buffer at any given time, and should be determined based on the characteristics of a particular system (e.g., I/O bandwidth, source latency)."); *id.* at 11:37-40 ("Play may halt for a variety of reasons. . . . control unit 114 may reach the . . . last frame in the global playback sequence."); *id.* at 5:63-65 ("Here, three clips are shown (Nc =3), each of a different length (N1, N2, N3) and stored in a different source. The three clips together form the global playback sequence."); *id.* at 11:63-12:2 ("[T]he current invention accesses

a global playback sequence consisting of several clips. The cliplist manager 116 specifies where each clip can be found (e.g., file name on the local disk, server identification number and file name), which frame in the clip is the first to play, and which frame in the clip is the last to play. The compilation of this data is known as the cliplist.").

The global playback sequence of three clips is shown in Figure 7 below:

FIG. 7



Accordingly, Hill teaches claim 1[g]. Further, Hill especially teaches claim 1[g] if WAG's proposed plain and ordinary meaning construction of the predetermined limitation is adopted. As discussed previously, WAG broadly contends that any specified data in the client buffer at any point (whether before the client connects to the server or after the client connects to the server) would cover this claim limitation. *See supra* 18-19 (WAG contending predetermined is a "specified number of media data elements" that "could be determined after connection to the network and after connection to the media source" and are not constant or the "same number").

Hill describes that the client buffer maintains a threshold of frames based on repeated requests for frames. Hill (Ex. 10), at 1:7-11, 3:6-11, 5:52-54, 6:58-67, 9:40-44, claims 8-9. Thus, Hill especially anticipates under WAG's broad plain and ordinary meaning construction given the client buffers specified data.

3. **Claim 4**

Claim 4 states, "4. The media player of claim 1, wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player."

Hill describes that the media player will receive the predetermined frames, *e.g.*, two frames, from the source at 48 frames per second, which is greater than the playback rate of 24 frames per second. Hill (Ex. 10), at 7:15-24 ("If the buffer manager 118 is more than one frame behind, **it will request two new frames**. However, in a preferred embodiment no more than two new frames will be requested for each request received from the cliplist manager 116 in order to avoid overloading the sources. Hence, **if the play rate is 24 frames per second, the sources will not be asked to supply more than 48 frames per second**. However, the present invention is **not limited to requesting only two frames at a time, as would be apparent to one skilled in the art**.") (emphasis added). Accordingly, Hill teaches claim 4.

To the extent it is argued that Hill does not teach claim 4, it is taught by Carmel. As discussed above, a POSITA would have been motivated to combine Hill and Carmel, Glaser, or TCP for claim 4.

Hill already describes the client receiving a predetermined number of frames. *See* claim 1[c] and [g]. Carmel teaches that the media player receives the media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player. *See supra* 43-46. Accordingly, Hill and Carmel teach claim 4.

A POSITA would have been motivated to combine Hill and Glaser or TCP for claim 4. Hill already describes the client receiving a predetermined number of segments. *See* claim 1[g]. Glaser or TCP transmits multimedia data faster than the playback rate. *See supra* 47. Accordingly, Hill and Glaser or TCP teach claim 4.

D. SNQP4: Shteyn and Hill render obvious claims 1 and 4

1. Motivation to Combine

A POSITA would have found it obvious to combine the teachings of Hill and Shteyn because the combination is merely a matter of applying a known technique to a known device for improvement to yield predictable results, including for claim limitation 1[g].

Shteyn and Hill both teach video distribution systems over the Internet, including requesting data by serial number and to repeat such requests. *See supra* 11-12; claim 1[g] below. A POSITA at the time of the purported invention was made would have been motivated to combine the similar systems and enhance Shteyn's functionality, including real-time playback. This includes, for instance, implementing Hill's buffer threshold. Hill (Ex. 10), at 6:58-67, 7:1-2, 7:12-24, 9:40-44, 10:55-60, claims 8-9.

A POSITA under *KSR* (*see supra* 20-21) would have been further motivated to implement such a function and would have had a reasonable expectation of success in doing so considering Shteyn's existing functionality. Shteyn already teaches buffering and real-time playback of sequential media segments. Shteyn (Ex. 4), at Abstract, at 1:62-2:3, 3:3-36, 3:63-4:3, Fig. 1. Moreover, this existing buffering (linked list) enables Shteyn to store the received predetermined data faster than the playback rate. *Id.* at Abstract, 1:62-66, 3:14-18, 3:31-36, 4:20-31. Thus, Shteyn could implement Hill's buffer threshold seamlessly.

Also, a POSITA would have been motivated to combine the references given the patentee's admissions that the TCP protocol was well-known for proper data delivery from server to client and that media players, including client buffers, were well-known for playback. *See supra* 2-3; *Unwired Planet*, 841 F.3d at 1003; *McCoy*, 850 F. App'x at 789. A POSITA would have been motivated to improve on this known knowledge, *e.g.*, transmitting data and storing such data in the client's buffer, by combining Hill's threshold buffering to further enhance Shteyn's buffering. *Advanced Micro Devices*, 848 F.2d at 1570; *WesternGeco*, 889 F.3d at 1329. Accordingly, a POSITA would have been motivated to combine Shteyn and Hill.

2. **Claim 1 [g]**

As discussed above, a POSITA would have been motivated to combine Shteyn and Hill for claim 1[g]. Hill teaches this claim limitation. *See supra* 59-61. Accordingly, Shteyn and Hill render obvious claims 1 and 4.

E. SNQP5: Feig anticipates or renders obvious with Carmel, Glaser, or TCP claims 1 and 4

1. **Motivation to Combine**

A POSITA would have found it obvious to combine the teachings of Carmel and Feig because the combination is merely a matter of applying a known technique to a known device for improvement to yield predictable results, including for claim 4.

Feig and Carmel disclose similar client-pull systems, including requesting data by serial number using a file listing of segments. *See supra* 12-13, 43-46; claim 1[c] below; WebPower Final Written Decision on Remand (Ex. 12), at 16-23. A POSITA under *KSR* (*see supra* 20-21) at the time of the purported invention was made would have been motivated to implement Carmel's sending faster than playback rate in Feig given the similar client-pull systems and to further

enhance Feig's functionality, including Feig streaming in a continuous or uninterrupted manner. Feig (Ex. 6), at 5:24-36, 5:44-53, Fig. 4.

A POSITA would have been further motivated to implement such function in Feig and would have had a reasonable expectation of success in doing so considering Feig's existing client and server functionality. For instance, Feig sends data at a high rate and implements client buffers that could receive data sent faster than the playback rate. Feig (Ex. 6), at 5:16-23, 6:21-38. As such, the predetermined segments received in Feig's player could be received faster than playback and properly buffered, resulting in enhanced real-time playback. Thus, Feig could implement Carmel's sending data faster than playback rate given Feig's existing server and client structure such that it could buffer data received faster than the playback rate.

Also, a POSITA would have been motivated to combine the references given the patentee's admissions that the TCP protocol was well-known for proper data delivery from server to client and that media players, including client buffers, were well-known for playback. *See supra* 2-3; *Unwired Planet*, 841 F.3d at 1003; *McCoy*, 850 F. App'x at 789. A POSITA would have been motivated to improve on this known knowledge, *e.g.*, transmitting data and storing such data in the client's buffer, by combining Carmel's transmitting faster than the playback rate to further enhance Feig's efficiency in buffering and playing streaming data. *Advanced Micro Devices*, 848 F.2d at 1570; *WesternGeco*, 889 F.3d at 1308. Accordingly, a POSITA would have been motivated to combine Feig and Carmel.

Additionally, a POSITA would have been motivated to combine Feig and Glaser. Feig and Glaser both teach video distribution systems over the Internet. *See supra* 12, 15; claim 1[a]-[c] below. Glaser transmits multimedia data faster than the playback rate to maintain a client buffer threshold. Glaser (Ex. 9), at 3:15-28, 3:31-36, 13:36-44, 16:2-10, 21:57-63, 22:41-59.

A POSITA under *KSR* (*see supra* 20-21) would have been motivated to implement Glaser's sending data faster than playback rate to further enhance Feig's playback in a continuous or uninterrupted manner. Feig (Ex. 6), at 5:24-36, 5:44-53, Fig. 4. Also, Glaser is cited on the front page of the '011 Patent, so a POSITA would have been motivated to review and apply it to other prior art, including Feig. A POSITA would have been further motivated to implement such function considering Feig's server structure and existing buffering ability discussed above. Feig already teaches buffering and real-time playback of sequential media segments. *Id.* at Abstract, 5:16-50. Thus, Feig could implement Glaser's sending data faster than playback seamlessly. Accordingly, a POSITA would have been motivated to combine Feig and Glaser.

2. **Claim 1**

(a) ***1[a] - A media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number, comprising***

Feig teaches a media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number.

Feig teaches that a server creates a file with a sequential list of URLs, each representing a video segment. Feig (Ex. 6), at Abstract ("The URLS consists of a header and a sequence of URLs. The method of the present invention receives the URLS data then sequentially accesses the data of each URL comprising the URLS . . . ."); *id.* at 6:18-20 ("The video data which will be streamed is partitioned into sequenced segments. A URLS is created, with the URL segment sequence as it's [sic: its] content."). The list of unique URLs is shown in Fig. 1 below:

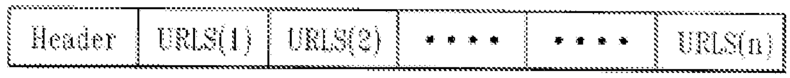


FIG. 1

Id. at Fig. 1; *id.* at 3:20-22 ("A URLS consists of a Header, containing a header file, and an ordered list URLS(j), where j is an index ranging from 1 to n."); *id.* at Fig. 2.

Each media segment in the URL file listing is associated with a serial number (URL or filename), as shown in Figure 1 above. *Id.* at 2:54-59 ("URLs may be a sequence of pre-existing files, or a new sequence of URL's may be created by partitioning a data resource such as video into contiguous time segments, storing each segment in an Internet location and creating a URL for each of these segments."); *id.* at 1:56-57 ("Each hot object in hypertext files is linked to a unique URL."); *id.* at 6:18-20 ("The video data which will be streamed is partitioned into sequenced segments. A URLS is created, with the URL segment sequence as it's [sic: its] content."); *id.* at 1:41-43 ("The URL complies with a draft standard for specifying an object on the Internet. It specifies the object format, the access method, and the location of the object."); *id.* at 1:62-65 ("Another situation arises when a sequence of video segments is located at various URLS, when linked together such sequences comprise one contiguous selection of video.")

Feig describes that the media player sequentially requests and plays each video segment by URL or serial number over the Internet. *Id.* at 2:60-61 ("The browser receives the URLS data, sequentially accesses the data of each URL comprising the URLS. . . ."); *id.* at 5:16-18 ("Next, the 'Fetch segment' module fetches the first segment and stores it in BUFF_A. This involves making requests for URL(1), URL(2), and so on, until URL(A1)."); *id.* at 4:1-5 ("In addition, the computer processing system includes a communication link 109, such as a network adapter or a modem coupled to the CPU 103, that allows the CPU 103 to communicate with other computer processing

systems over the communication link, for example over the Internet."); *id.* at 5:47-50 ("If the contents of the URLs comprise a video sequence, then the video will be displayed from beginning to end in a continuous manner as is typical when using a streaming server and a stream capable player."); *id.* at Abstract ("The method of the present invention receives the URLS data then sequentially accesses the data of each URL comprising the URLS . . . and times the calls for subsequent URLs in the sequence accordingly so that the arrival of the linked data nearly simulates actual streaming."); *id.* at 1:6-11 ("The present invention relates to Internet and Intranet Browsers, more specifically, to Browsers equipped with functionality to process a sequence of URL requests automatically."); *id.* at 6:18-20 ("The video data which will be streamed is partitioned into sequenced segments. A URLS is created, with the URL segment sequence as it's [sic: its] content."); *id.* at 2:1-3.

Accordingly, Feig teaches claim 1[a]. Further, Feig teaches 1[a] if WAG's proposed plain and ordinary meaning construction of "serial number / identifier" is adopted As discussed previously, WAG contends that "serial number/identifier" is not limited to "consecutive [numbers] or limited to numerals," and thus can broadly encompass several forms. *See supra* 18. Feig teaches that the requested URLs or segments are not limited to any particular form and can even take several forms, including an "ordered list" or video sequence that will be displayed from "beginning to end in a continuous manner." Feig (Ex. 6), at 3:20-22 ("A URLS consists of a Header, containing a header file, and an ordered list URLS(j), where j is an index ranging from 1 to n."), 5:47-50 ("If the contents of the URLs comprise a video sequence, then the video will be displayed from beginning to end in a continuous manner as is typical when using a streaming server and a stream capable player.") Thus, Feig especially anticipates under WAG's broad plain and ordinary meaning construction. *Id.* at 1:39-43, 1:56-57, 2:60-67, 5:16-23, Abstract.

(b) ***1[b] - a processor; a memory; a connection to the network; and***

Feig teaches a processor, memory, and connection to the network. Specifically, Feig describes that the "invention may be implemented on any computer processing system including, for example, a personal computer or a workstation." Feig (Ex. 6), at 3:38-40. This computer contains "memory 101, one or more central processing units (CPU) 103, and one or more user input devices 107 . . . a nonvolatile memory such as ROM and other nonvolatile storage devices 108 such as a fixed disk drives that store operating systems and application programs that are loaded into the memory 101 for execution by the CPU 103." *Id.* at 3:42-51; *id.* at 3:51-67 (describing other components of computer). This computer also includes "a communication link 109, such as a network adapter or a modem coupled to the CPU 103, that allows the CPU 103 to communicate with other computer processing systems … over the Internet." *Id.* at 4:1-5. Figure 3 below shows the processor, memory, and communication link:

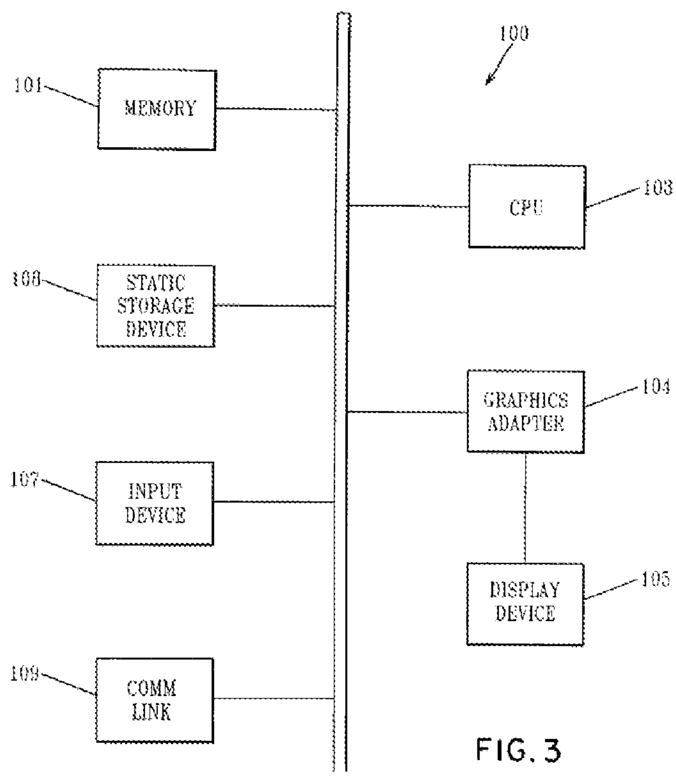


FIG. 3

Id. at Fig. 3.

Accordingly, Feig teaches claim 1[b].

(c) ***1[c] - media player software comprising instructions to cause the media player to request from the media source a predetermined number of media data elements;***

Feig teaches media player requesting from the media source a predetermined number of media data elements.

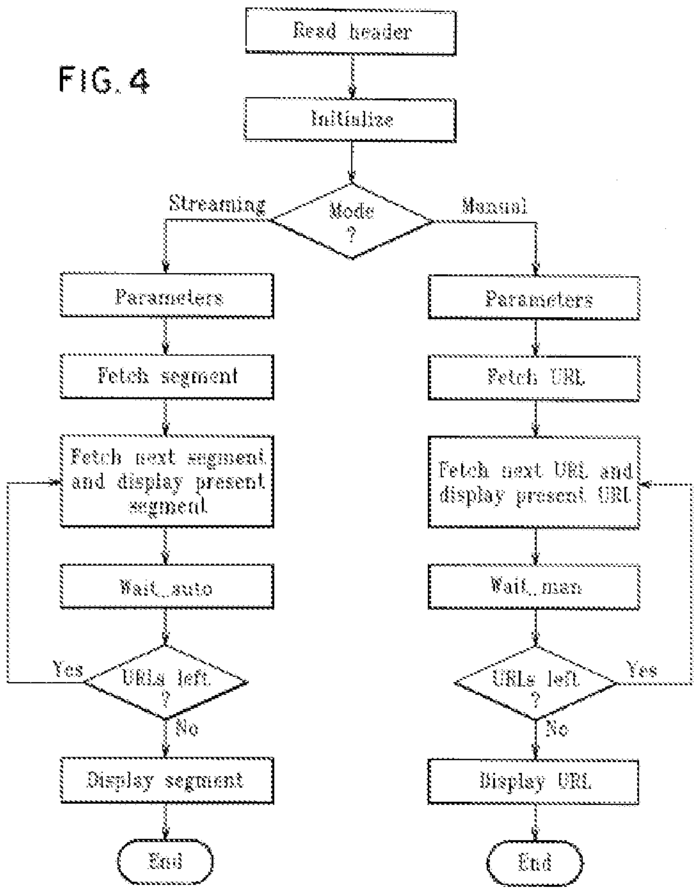
Feig teaches that the client requests a predetermined number of media data elements by fetching from the server and storing one segment in a buffer (BUFF_A) and thereafter fetching and storing another segment in a separate buffer (BUFF_B). Feig (Ex. 6), at 5:16-23 ("Next, the '**Fetch segment' module fetches the first segment** and stores it in **BUFF_A**. This involves making **requests for URL(1), URL(2), and so on, until URL(A1)**. As soon as all the data for the first segment arrives, the **'**Fetch next segment-and display present-segment' module begins **fetching the second segment and storing it in BUFF_B** and simultaneously displaying the decoded contents from the first segment stored in BUFF_A.") (emphasis added); *id.* at 5:30-36 ("If there are [segments remaining], the control is passed back to the 'Fetch next segment-and display present-segment' module, which initiates a fetch of a next segment, and stores it in the alternate buffer, BUFF_A for segment three. At the same time it starts displaying the decoded output from the current segment, which is already stored in BUFF_B.").

Feig will repeatedly request two segments (one after another) and store such segments in alternating two buffers, BUF_A and BUF_B, for the duration of the program. *Id.* at 5:37-43 ("This process is continued, with one segment being stored in one buffer and the previous segment being played back from the other, until the 'URLs left?' decision module determines that there are no

more segments left to fetch, Then, the last segment is displayed by the 'Display segment' module and the streaming process terminates with the 'End' module.").

The predetermined requesting of segments for the duration of the program is shown in Figure 4 below, particularly "Fetch segment," "Fetch next segment and display present segment," and "URLs left?":

FIG. 4



Id. at Fig. 4.

Accordingly, Feig teaches claim 1[c]. Further, Feig especially teaches claim 1[c] if WAG's construction of "predetermined" discussed with reference to claim 1[g] as any specified data is applied here. *See supra* 18-19. Feig describes requesting predetermined segments and storing such segments in alternating buffers (BUF_A and BUF_B) for the duration of the program. *See*

above. Thus, Feig especially anticipates under WAG's broad plain and ordinary meaning construction given specified data is requested.

(d) ***1[d] - instructions to cause the media player to receive media data elements sent to the media player by the media source and store the media data elements in the memory;***

Feig teaches the media player to receive media data elements sent to the media player by the media source and store the media data elements in the memory.

As discussed, Feig teaches receiving and buffering segments in memory, BUFF_A and BUFF_B. *See* claim 1[c]; *see also* Feig (Ex. 6), at 5:44-50 ("[I]f the streaming criteria are met, then the entire content of all the URLs comprising the URLS will be displayed in proper timing order as prescribed by the content creator. If the contents of the URLs comprise a video sequence, then the video will be displayed from beginning to end in a continuous manner as is typical when using a streaming server and a stream capable player."); *id.* at 5:28-36 ("When both fetch and display are done, the 'URL's left?' decision module determines whether segments are remaining to be fetched. If there are, the control is passed back to the 'Fetch next segment-and display present-segment' module, which initiates a fetch of a next segment, and stores it in the alternate buffer, BUFF_A for segment three. At the same time it starts displaying the decoded output from the current segment, which is already stored in BUFF_B."); *id.* at Fig. 4.

Feig provides an example of the player receiving and buffering such media segments for uninterrupted playback:

> A 60 minute MPEG encoded video stream is partitioned into one minute segments. The data rate of the encoded video is 1.5 Mbps, each one minute segment is therefore encoded with 90 megabits, and the entire video is encoded with 5.4 Gigabits (or 675 MBYTES) of data. The bandwidth from the server to the browser is 3 Mbps. The browser determines that the time delay between its request for data and the beginning of the arrival of the data is less than 20 seconds. The time to deliver a one minute segment of

> video over the channel is 30 seconds. Thus, the total time from initiating the request for a minute segment till the entire data for that segment arrives is less than 50 seconds, and certainly less than one minute, which is the playing time of one segment. Hence the streaming criteria is satisfied. The browser allocates 2 or more memory buffers, each of 12 megabytes, for temporarily storing the incoming video data, and utilizes them in an alternating fashion.

Id. at 6:21-38.

Accordingly, Feig teaches claim 1[d].

(e) ***1[e] - instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer;***

Feig teaches the player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer.

Feig teaches a player buffer manager of operating system and/or application programs (*e.g.*, browser) to "perform the methods of the present invention" including managing the requesting and buffering segments by serial number described in claim 1[a] and 1[c]. Feig (Ex. 6), at 4:10-15 ("Application programs and the operating system executed by the CPU 103 may perform the methods of the present invention described below. Alternatively, portions or all of the methods described below may be embodied in hardware that works in conjunction with the application program and the operating system executed by the CPU 103."); *id.* at 3:49-50 ("operating systems and application programs that are loaded into the memory 101. . . ."); *see also* Fig. 3 (showing components). For instance, Feig teaches that the browser or buffer manager will allocate two buffers for storing the requested media segments. *Id.* at 6:34-38 ("browser allocates 2 or more

memory buffers, each of 12 megabytes, for temporarily storing the incoming video data, and utilizes them in an alternating fashion.").

Feig further maintains a record of the serial number of the last media data element that has been received and stored in the player buffer because it maintains a record of each segment by serial number in its buffers.

As described, Feig sequentially requests each segment by serial number (URL or filename) from the URL file. *See* claim 1[a] and 1[c]. The buffer manager (operating system and/or application programs such as a browser) will store each requested segment with the serial number in alternating two buffers (BUFF_A and BUFF_B). *Id.* at 6:34-38 ("browser allocates 2 or more memory buffers, each of 12 megabytes, for temporarily storing the incoming video data, and utilizes them in an alternating fashion."); *id.* at 5:16-23 ("Next, the 'Fetch segment' module fetches the first segment and stores it in BUFF_A. This involves making requests for URL(1), URL(2), and so on, until URL(A1). As soon as all the data for the first segment arrives, the 'Fetch next segment-and display present-segment' module begins fetching the second segment and storing it in BUFF_B and simultaneously displaying the decoded contents from the first segment stored in BUFF_A."); *id.* at Claim 11 (". . . d) partitioning the URLs comprising the URLS into segments . . . allocating at least two buffers each of size BUFF; f) fetching the first segment and storing it in one of the empty buffers; g) fetching the next segment and storing it in an empty buffer, and simultaneously displaying the segment stored in a full buffer until no more segments remain to be fetched . . . ."); *id.* at 5:24-36, Fig. 4.

This buffering or recording of the last media data element in the two buffers continues until the end of the program. *Id.* at 5:37-43 ("This process is continued, with one segment being stored in one buffer and the previous segment being played back from the other, until the 'URLs left?'

decision module determines that there are no more segments left to fetch, Then, the last segment is displayed by the 'Display segment' module and the streaming process terminates with the 'End' module.").

Thus, Feig's operating system and/or application programs (*e.g.*, browser) teach maintaining a record of the serial number of the last data element that has been received because it stores or records each unique segment from the URL list in the two buffers, including the last one received. Accordingly, Feig teaches claim 1[e].

(f) ***1[f] - instructions to cause the media player to play media data elements sequentially from the player buffer; and***

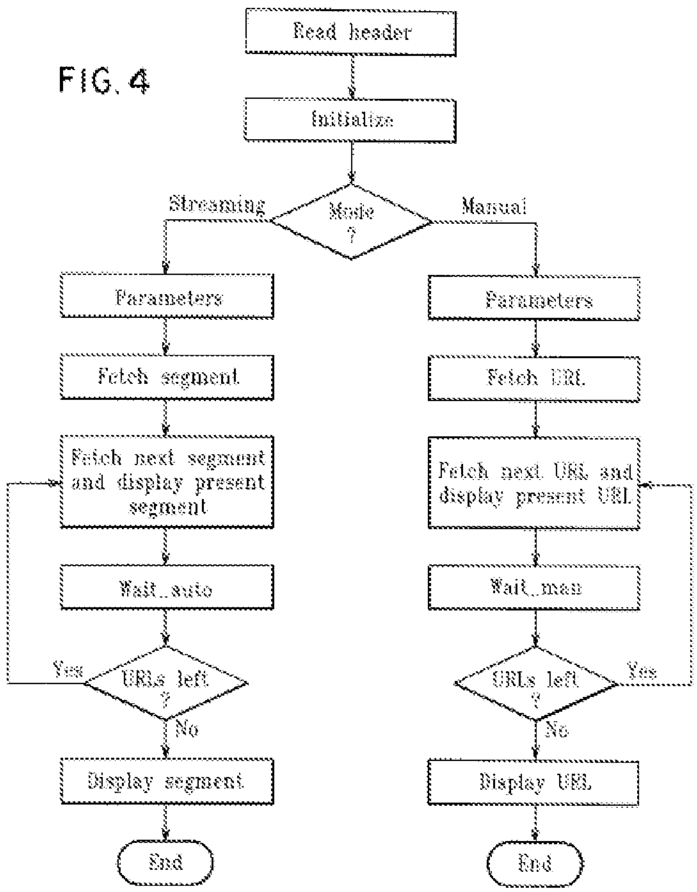
Feig teaches that the media player to play media data elements sequentially from the player buffer.

Feig describes that the segments are in a sequential or continuous manner and such segments will be played back in the same manner from the buffers. Feig (Ex. 6), at 5:47-50 ("If the contents of the URLs comprise a video sequence, then the video will be displayed from beginning to end in a continuous manner as is typical when using a streaming server and a stream capable player."); *id.* at 2:60-67 ("The browser receives the URLS data, sequentially accesses the data of each URL . . . ."); *id.* at 5:16-18 ("Next, the 'Fetch segment' module fetches the first segment . . . . This involves making requests for URL(1), URL(2), and so on, until URL(A1)."); *id.* at 6:18-20 ("The video data which will be streamed is partitioned into sequenced segments. A URLS is created, with the URL segment sequence as it's [sic: its] content."); *id.* at claim 2 ("A method of claim 1, further comprising dividing the resource into a plurality of segments and associating each segment with one of said URLS."); *id.* at 1:56-57 ("Each hot object in hypertext files is linked to a unique URL."); *id.* at 2:54-59 ("URLs may be a sequence of pre-existing files,

or a new sequence of URL's may be created by partitioning a data resource such as video into contiguous time segments, storing each segment in an Internet location and creating a URL for each of these segments"). *id.* at 1:62-2:3.

The displaying or playing of sequential segments is shown below in Figure 4:

FIG. 4



Id. at Fig. 4; *id.* at 5:37-43 ("This process is continued, with one segment being stored in one buffer and the previous segment being played back from the other, until the 'URLs left?' decision module determines that there are no more segments left to fetch, Then, the last segment is displayed by the 'Display segment' module and the streaming process terminates with the 'End' module.")

Accordingly, Feig teaches claim 1[f].

(g) ***1[g] - instructions to cause the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.***

Feig teaches that the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.

Feig describes transmitting a request by serial number from the URL listing to the server for a media segment and repeating such requests by serial number for the other sequential segments for the duration of program. *See* claim 1[a], [c], and [e]; *see also* Feig (Ex. 6), at Abstract, 2:60-67 ("The browser receives the URLS data, sequentially accesses the data of each URL comprising the URLS. . . ."); 2:60-67, 5:16-23, 5:30-36, 5:37-43, 5:47-53, claim 11, Fig. 4.

Feig repeats these requests to maintain a predetermined number of media data elements in the player buffer until the last media data element comprising the program has been received because Feig repeatedly requests two segments (one after another) and stores them in alternating two buffers, BUFF_A and BUFF_B. *See* claim 1[c]; Feig (Ex. 6), at 5:16-23 ("Next, the **'Fetch segment' module fetches the first segment and stores it in BUFF_A. This involves making requests for URL(1), URL(2), and so on, until URL(A1). As soon as all the data for the first segment arrives, the 'Fetch next segment-and display present-segment' module begins fetching the second segment and storing it in BUFF_B** and simultaneously displaying the decoded contents from the first segment stored in BUFF_A.") (emphasis added); *id.* at 5:30-36

("If there are [segments remaining], the control is passed back to the 'Fetch next segment-and display present-segment' module, which initiates a fetch of a next segment, and stores it in the alternate buffer, BUFF_A for segment three. At the same time it starts displaying the decoded output from the current segment, which is already stored in BUFF_B."); *id.* at Fig. 4.

The storing of segments in the predetermined two buffers is continued until the end of the program or video as shown in Figure 4 above. *Id.* at Fig. 4 ("URLs left?"); *id.* at 5:47-50 ("If the contents of the URLs comprise a video sequence, then the video will be displayed from beginning to end in a continuous manner as is typical when using a streaming server and a stream capable player."); *id.* at 5:37-43 ("This process is continued, with one segment being stored in one buffer and the previous segment being played back from the other, until the 'URLs left?' decision module determines that there are no more segments left to fetch, Then, the last segment is displayed by the 'Display segment' module and the streaming process terminates with the 'End' module."); *id.* at 5:16-18 ("Next, the 'Fetch segment' module fetches the first segment and stores it in BUFF_A. This involves making requests for URL(1), URL(2), and so on, until URL(A1). . . ."); *id.* at Abstract ("The method of the present invention receives the URLS data then sequentially accesses the data of each URL comprising the URLS . . . and times the calls for subsequent URLs in the sequence accordingly so that the arrival of the linked data nearly simulates actual streaming.")

Accordingly, Feig teaches claim 1[g]. Further, Feig especially teaches claim 1[g] if WAG's proposed plain and ordinary meaning construction of the predetermined limitation is adopted. As discussed previously, WAG broadly contends that any specified data in the client buffer at any point (whether before the client connects to the server or after the client connects to the server) would cover this claim limitation. *See supra* 18-19 (WAG contending predetermined

is a "specified number of media data elements" that "could be determined after connection to the network and after connection to the media source" and are not constant or the "same number").

Feig maintains two predetermined buffers, BUFF_A and BUFF_B, of segments for the entire program. Feig (Ex. 6), at 5:16-23, 5:37-43, 5:47-50, Fig. 4. Thus, Feig especially anticipates under WAG's broad plain and ordinary meaning construction given the client buffers specified data.

3. **Claim 4**

Claim 4 states, "4. The media player of claim 1, wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player."

Feig teaches the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player.

As described, Feig requests and receives a stream of predetermined media segments. *See* claim 1[g]. Feig further describes the media player receiving the requested segments faster than the playback rate and storing such segments in the two allocated buffers. Specifically, Feig teaches that a "**60 minute MPEG encoded video stream** is partitioned into **one minute segments**. The data rate of the **encoded video is 1.5 Mbps**, each one minute segment is therefore encoded with 90 megabits, and the entire video is encoded with 5.4 Gigabits (or 675 MBYTES) of data." Feig (Ex. 6), at 6:21-26 (emphasis added). Feig further describes that the "**bandwidth from the server**

to the browser is 3 Mbps." *Id.* at 6:26-27 (emphasis added). Thus, the transmission rate (3 Mbps) is greater than the data rate of the encoded segment to be played back (1.5 Mbps).

Feig then states that the, "browser determines that the time delay between its request for data and the beginning of the arrival of the data is less than 20 seconds . . . [and] time to deliver a one minute segment of video over the channel is 30 seconds." *Id.* at 6:27-31. "Thus, the total time from initiating the request for a minute segment till the entire data for that **segment arrives is less than 50 seconds**, and **certainly less than one minute, which is the playing time of one segment**." *Id.* at 6:31-34 (emphasis added). Feig concludes the example by noting that "the streaming criteria is satisfied . . . [and] [t]he **browser allocates 2 or more memory buffers, each of 12 megabytes, for temporarily storing the incoming video data, and utilizes them in an alternating fashion**." *Id.* at 6:34-38 (emphasis added); *id.* at 6:18-20 ("The video data which will be streamed is partitioned into sequenced segments. A URLS is created, with the URL segment sequence as it's [sic: its] content. As an illustration, please consider the following.")

In sum, given the transmission rate of each segment (3 Mbps) is greater than the encoded rate of the video (1.5 Mbps) and given each segment arrives at the client in less time (below 50 seconds) than the playback time (1 minute), Feig teaches claim 4 or receiving the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player.

To the extent it is argued that Feig does not disclose claim 4, it is taught by Carmel. As discussed above, a POSITA would have been motivated to combine Feig and Carmel for claim 4. Feig already describes the client receiving a predetermined number of segments. *See* claim 1[g]. Carmel teaches that the media player receives the media data elements at a rate more rapid than

the rate at which the media data elements are to be played out by the media player. *See supra* 43-46. Accordingly, Feig and Carmel teach claim 4.

Additionally, Feig and Glaser or TCP teach claim 4. A POSITA would have been motivated to combine Feig and Glaser or TCP for claim 4. Feig already describes the client receiving a predetermined number of segments. *See* claim 1[g]. Glaser or TCP teaches sending data at a rate faster than the playback rate. *See supra* 47. Accordingly, Feig and Glaser or TCP teach claim 4.

F. SNQP6: Carmel anticipates or renders obvious with Hill claims 1 and 4

1. Motivation to Combine

A POSITA would have found it obvious to combine the teachings of Carmel and Hill because the combination is merely a matter of applying a known technique to a known device for improvement to yield predictable results, including for claim 1[g].

Carmel and Hill both teach video distribution systems over the Internet, including requesting data by serial number and to repeat such requests. *See supra* 11-13; WebPower Final Written Decision on Remand (Ex. 12), at 16-23. Further, both maintain a record of the last serial number received. *See supra* 11-13. A POSITA under *KSR* (*see supra* 20-21) at the time of the purported invention was made would have been motivated to combine the similar systems. For instance, a POSITA would have been motivated to implement Hill's threshold of buffering (Hill (Ex. 10), at 6:58-67, 7:1-2, 9:40-44, 10:55-60, claims 8-9) to further enhance Carmel's real-time or synchronized playback. Carmel (Ex. 3), at 2:1-21, 7:36-40, 10:50-54.

A POSITA would have been further motivated to implement such function and would have had a reasonable expectation of success in doing so considering Carmel's existing functionality. Carmel already teaches decoding, buffering, and playing sequential media slices (*id.* at 7:4-

17,7:59-8:5, 10:25-48, claim 10, Fig. 6A) as well as maintaining a record of the serial number of the last data element received (*id.* at 2:1-21, 8:18-26, 8:32-36, 10:25-48, Figs. 3, 6A). Thus, Carmel could implement Hill's buffer threshold seamlessly.

Also, a POSITA would have been motivated to combine the references given the patentee's admissions that the TCP protocol was well-known for proper data delivery from server to client and that media players, including client buffers, were well-known for playback. *See supra* 2-3; *Unwired Planet*, 841 F.3d at 1003; *McCoy*, 850 F. App'x at 789. A POSITA would have been motivated to improve on this known knowledge, *e.g.*, transmitting data and storing such data in the client's buffer, by combining Hill's threshold buffering to further enhance Carmel's downloading and playing of slices. *Advanced Micro Devices*, 848 F.2d at 1570; *WesternGeco*, 889 F.3d at 1329. Accordingly, a POSITA would have been motivated to combine Carmel and Hill.

2. **Claim 1**

(a) ***1[a] - A media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number, comprising***

Carmel teaches a media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number.

Carmel teaches the client's browser and/or media player downloading and playing streaming media consisting of slices. Carmel (Ex. 3), at 7:13-17 ("because HTTP is supported by substantially all modern Web browsers, clients 30 will typically need only add a Java applet or

plug-in to their existing Web browsers, as described further hereinbelow, in order to receive and play back the broadcast."); *id.* 10:28-35 ("The operation of client is controlled by a Java applet, which may be downloaded from server 36, and includes facilities for carrying out the steps shown in FIG. 6A . . . . A sample applet of this sort is incorporated herein in the software appendix, as further described hereinbelow."); *id.* at 3:66-4:1 ("Preferably, the one or more client computers decode the sequence and play back the data stream responsive to the indices of the files . . . ."); *id.* at 6:38-40 ("Computer 34 and clients 30 preferably comprise conventional personal computers or workstations."), 10:36-54; *id.* at Figs. 2, 6A; *id.* at 10:24-27 ("FIG. 6A is a flow chart illustrating the operation of clients 30 in downloading and playing back data stream 40 (FIG. 3A) . . . .").

Carmel further teaches that the media player receives the media data over the Internet from the server. *Id.* at 2:1-15 ("a transmitting computer generates a data stream and broadcasts the data stream via a network server to a plurality of clients . . . . The clients download the data stream from the server, preferably using an Internet protocol, as well, most preferably the Hypertext Transfer Protocol (HTTP), or alternatively, using other protocols, such as UDP or RTP [Real Time Protocol], which are similarly known in the art."); *id.* at 7:5-9 ("Clients 30 preferably download the sequence using the Hypertext Transfer Protocol (HTTP), although other Internet protocols may also be used, such as UDP or RTP, as noted hereinabove with reference to uploading by computer 34.")

This distribution of streaming media over the Internet to multiple users is shown in Figure 2 below:

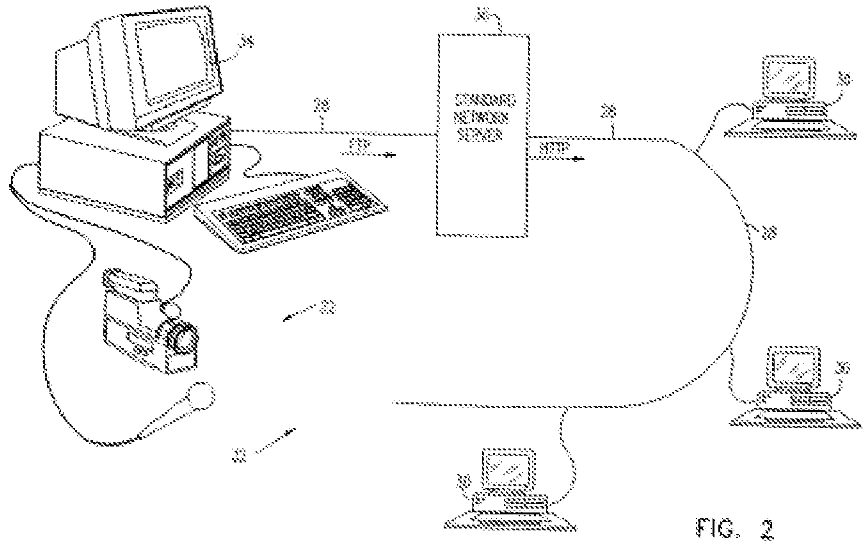
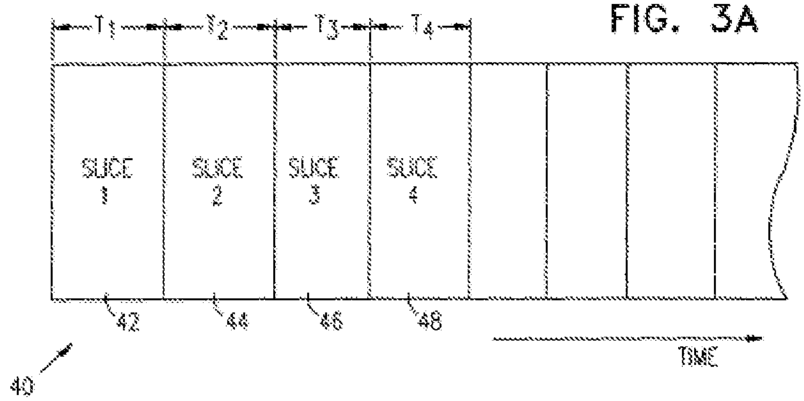


FIG. 2

Id. at Fig. 2; *see also id.* at 1:16-22 ("In network broadcasting, data are transmitted over a network in real time from a single transmitting computer to a plurality of clients simultaneously. The network may be a LAN, a WAN, an intranet or a public network such as the Internet. Network broadcasting is most commonly used to stream multimedia data, typically comprising images and sound."); *id.* at 5:19-32, 6:28-35, 6:50-56; claims 2, 27-28, 38-39, 41.

Carmel describes that each of the media data elements or slices is associated with a serial number. Carmel describes that a "data stream is divided into a sequence of segments or slices of the data, preferably time slices, wherein the data are preferably compressed. Each slice is preferably assigned a respective slice index." *Id.* at 2:4-7; *id.* at 9:66-10:1 ("The data are compressed at step 80, and are then 'sliced' at step 82 into files 42, 44, 46, 48, etc., as shown in FIG. 3A."). Carmel also provides an example of sequential media segments or slices: "Data stream 40 comprises a series of data slices 42, 44, 46, 48, etc. Each slice contains a segment of video and/or audio data, corresponding to a respective, successive time interval labeled T1, T2, T3, etc…." *Id.* at 7:22-26. The sequential slices are shown in Figure 3A below.

FIG. 3A



Id. at Fig. 3a; *see also id.* at 3:24-34, 8:47-50.

Accordingly, Carmel teaches claim 1[a]. Further, Carmel especially teaches claim 1[a] if WAG's proposed plain and ordinary meaning construction of "serial number / identifier" is adopted. As discussed previously, WAG contends that "serial number/identifier" is not limited to "consecutive [numbers] or limited to numerals," and thus can broadly encompass several forms. *See supra* 18. Carmel teaches that the slice ID is not limited to a specific form and indeed can include a numeral and/or consecutive numbers. Carmel (Ex. 3), at 7:59-8:5, Fig. 3A. Thus, Carmel especially anticipates under WAG's broad plain and ordinary meaning construction.

(b) ***1[b] - a processor; a memory; a connection to the network; and***

Carmel teaches a processor, memory, and connection to the network. Carmel teaches that "Computer 34 and clients 30 preferably comprise conventional personal computers or workstations." Carmel (Ex. 3), at 6:38-40. Such a personal computer or workstation have processors and memory. This memory in the computer is further shown by the client downloading slices and displaying such slices in the form of a graphical slider. *Id.* at 8:18-26 ("FIG. 3C is a schematic representation of a user interface graphic 'slider' 55, available to users of computers 30 . . . . Slider 55, which is preferably displayed on the screens of computers 30, includes a bar 56 and a movable indicator 58. The symbols J, J+1, J+2, . . . N in the figure are the indices of the

slices of stream 40 that are stored on server 36, wherein N is the index of the most recent slice, and J is the index of the earliest stored slice."); 10:42-48 ("The client first reads index file 50 (FIG. 3B), and graphic 56 (FIG. 3C) is displayed by the client, so that a user can decide and indicate at which slice of data stream 40 to begin downloading. Responsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc."); *id.* at 8:32-36 ("When one of computers 30 reads index file 50 and begins to download stream 40, indicator 58 preferably marks the most recent slice, as shown in FIG. 3C. This is the point at which the download will begin, unless the user of the computer chooses otherwise.").

These user computers, with memory, are shown in Figure 2 below:

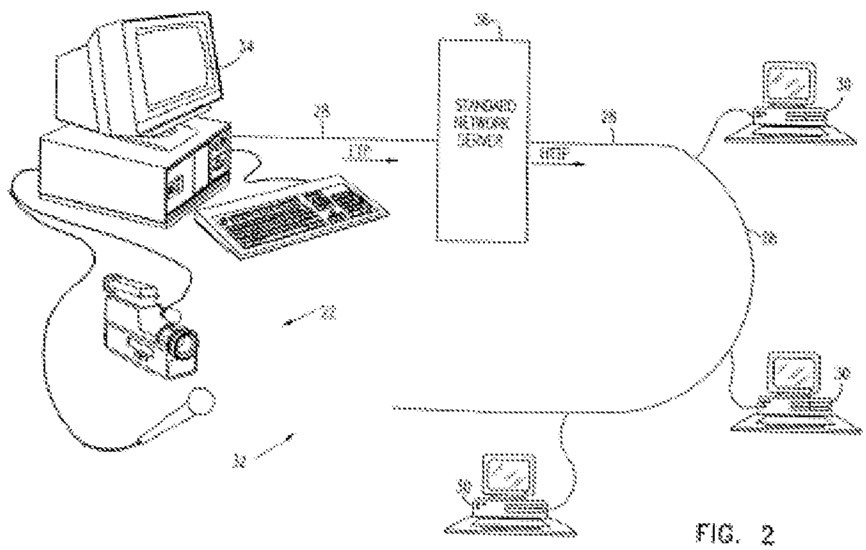


FIG. 2

Id. at Fig. 2.

Carmel further teaches a connection to the network as described in Figure 2 above and in claim 1[a]. *See also Id.* at 2:11-15 ("The clients download the data stream from the server, preferably using an Internet protocol, as well, most preferably the Hypertext Transfer Protocol (HTTP), or alternatively, using other protocols, such as UDP or RTP [Real Time Protocol], which

are similarly known in the art."); *id.* at 7:5-9 ("Clients 30 preferably download the sequence using the Hypertext Transfer Protocol (HTTP), although other Internet protocols may also be used, such as UDP or RTP, as noted hereinabove with reference to uploading by computer 34.")

Accordingly, Carmel teaches claim 1[b].

(c) ***1[c] - media player software comprising instructions to cause the media player to request from the media source a predetermined number of media data elements;***

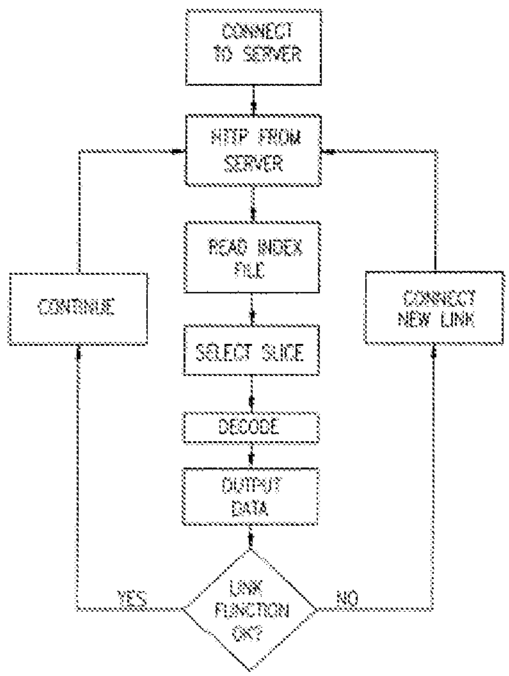
Carmel teaches that the media player requests from the media source a predetermined number of media data elements.

Carmel teaches requesting from the server a predetermined number of media data elements given Carmel downloads a specified number of slices (*e.g.*, one, two, or five slices) from the index file by identifier and stores such slices. Carmel (Ex. 3), at 10:36-40 ("Each client 30 connects to server 36, optionally using multiple HTTP links . . . . Typically, client 30 opens one or two HTTP links, over which files 42, 44, 46, etc., are downloaded in successive alternation . . . ."); *id.* at 9:20-31 ("Preferably, five links 60, 62, 64, 66 and 68 are opened and operate simultaneously over a single modem line. . . . Files 42, 44, 46, 48, etc., in stream 40 are transmitted respectively over links 60, 62, 64, 66 and 68 in successive alternation . . . . Alternatively, more than five links may be opened, so that more than five files may accordingly be transmitted in parallel."); *id.* at 10:42-48 ("The client first reads index file 50 (FIG. 3B), and graphic 56 (FIG. 3C) is displayed by the client, so that a user can decide and indicate at which slice of data stream 40 to begin downloading. Responsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc."); *id.* at 11:1-5 ("each client 30 connects to the server, generally using a single HTTP link. After reading header 43 and, preferably, making an

initial assessment of the link bandwidth, the client selects one of the available quality levels in the stream."); *id.* at Fig. 3C (shown further below).

This downloading of slices is shown in Figures 6A ("Read Index File" and "Select Slice") below.

FIG. 6A



Id. at Fig. 6A; *id.* at Fig. 6B; *id.* at 8:1-5 ("When one of computers 30 connects to server 36 and begins to download the data stream, it first reads the index file in order to identify at what point in stream 40 to begin and to start receiving the data stream . . . ."); Claim 10 ("10. A method according to claim 9, wherein downloading the sequence comprises selecting a file in the sequence earlier than the file whose index is contained in the index file and downloading at least a portion of the encoded sequence of files beginning with the selected file.").

Accordingly, Carmel teaches claim 1[c]. Further, Carmel especially teaches claim 1[c] if WAG's construction of "predetermined" discussed with reference to claim 1[g] as any specified data is applied here. *See supra* 18-19. Carmel requests a specified number of slices. *See* above. Thus, Carmel especially anticipates under WAG's broad plain and ordinary meaning construction given specified data is requested.

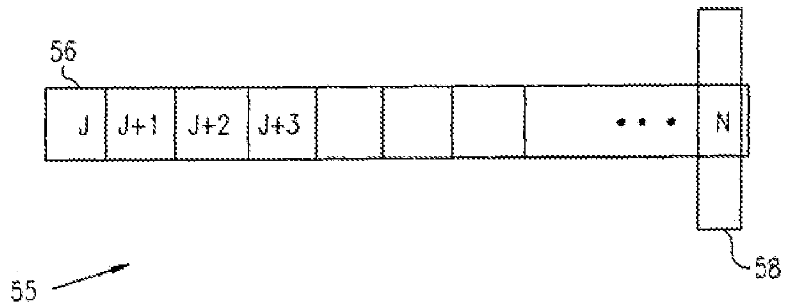
(d) ***1[d] - instructions to cause the media player to receive media data elements sent to the media player by the media source and store the media data elements in the memory;***

Carmel teaches that the media player receives the media data elements sent to the media player by the media source and store the media data elements in the memory.

As discussed, Carmel describes that the media player stores downloaded slices by serial number and then decodes and plays such slices. *See* claim 1[a] and [c]; *see also* Carmel (Ex. 3), at 10:45-54 ("Responsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc. In the case of a multimedia stream, client 30 reconstructs and outputs the multimedia data for the appreciation of a user. Time stamps in the data stream are used to synchronize the data, so that the multimedia sequence is played back just as it was input at computer 34, preferably with only a minimal necessary transmission and decoding delay."); *id.* at 10:64-11:8 ("FIG. 6B is a flow chart illustrating the operation of clients 30 in downloading and playing back multi-level data stream 41 (FIG. 3D) transmitted from server 36 . . . . Responsive to the selection, server 36 begins to transmit data slices at the chosen quality level. The slices are received, decoded and output by the client."); *id.* at 7:36-40.

Carmel also teaches displaying the stored slices to the user in a graphical slider shown below in Figure 3C:

FIG. 3C



Id. at Fig. 3C; *id.* at 8:32-36 ("When one of computers 30 reads index file 50 and begins to download stream 40, indicator 58 preferably marks the most recent slice, as shown in FIG. 3C. This is the point at which the download will begin, unless the user of the computer chooses otherwise."); *id.* at 8:21-28 ("Slider 55, which is preferably displayed on the screens of computers 30, includes a bar 56 and a movable indicator 58. The symbols J, J+1, J+2, . . . N in the figure are the indices of the slices of stream 40 that are stored on server 36, wherein N is the index of the most recent slice, and J is the index of the earliest stored slice. J may indicate the first slice in the sequence, if all of the files are stored on server 36 . . . ."). Accordingly, Carmel teaches claim 1[d].

(e) ***1[e] - instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer;***

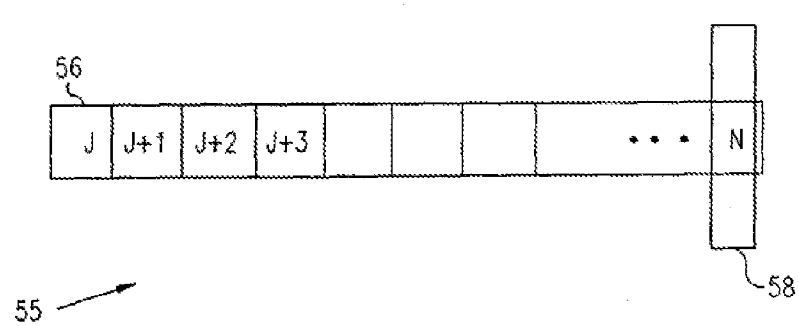
Carmel teaches a buffer manager consisting of a Java applet and/or browser, which manages or controls the player buffer in memory. This includes the downloading, decoding, and playing of slices. *See* claim 1[b] (describing memory); Carmel (Ex. 3), at 10:25-35 ("FIG. 6A is a flow chart illustrating the operation of clients 30 in downloading and playing back data stream 40 (FIG. 3A) transmitted by computer 34, in accordance with a preferred embodiment of the present invention. **The operation of client is controlled by a Java applet** . . . and **includes**

facilities for carrying out the steps shown in FIG. 6A, as well as for error detection and, optionally, correction in communications received by the clients and for other functions known in the art. A **sample applet of this sort is incorporated herein in the software appendix**, as further described hereinbelow.") (emphasis added); *id.* at Fig. 6A ("Select Slice," "Decode," "Output Data"); *id.* at 7:4-17 ("Clients 30 connect to server 36 and receive the multimedia sequence, substantially in real time. Clients 30 preferably download the sequence using the Hypertext Transfer Protocol (HTTP), although other Internet protocols may also be used, such as UDP or RTP . . . . because HTTP is supported by substantially all modern Web browsers, clients 30 will typically need only add a Java applet or plug-in to their existing Web browsers, as described further hereinbelow, in order to receive and play back the broadcast.")

Carmel's client or Java applet maintains a record of the serial number of the last data element received in the buffers, including the client downloading and maintaining numbered slices "42, 44, and 46, etc." *Id.* at 10:27-50 ("The operation of client is controlled by a Java applet . . . . The client first reads index file 50 (FIG. 3B), and graphic 56 (FIG. 3C) is displayed by the client, so that a user can decide and indicate at which slice of data stream 40 to begin downloading. Responsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc. In the case of a multimedia stream, client 30 reconstructs and outputs the multimedia data for the appreciation of a user."); *id.* at Fig. 6A ("Select Slice," "Decode," "Output Data"). Thus, Carmel stores a record of the last slice downloaded (*e.g.*, "46"). The client also tracks the last slice downloaded for proper playback. *Id.* at 2:17-21 ("The division of the data stream into slices and the inclusion of the slice indices in the data stream to be used by the clients in maintaining synchronization allows the broadcast to go on substantially in real time without the use of special-purpose hardware."); *id.* at 7:37-40.

Additionally, and separately, Carmel maintain a record of the serial number of the last media data element that has been received and stored in the player buffer because it teaches a graphical slider on the client that records which slices were downloaded, including the last one or "most recent slice," and which slices to download next. *Id.* at 8:32-36 ("When one of computers 30 reads index file 50 and begins to download stream 40, **indicator 58 preferably marks the most recent slice, as shown in FIG. 3C**. **This is the point at which the download will begin**, unless the user of the computer chooses otherwise.") (emphasis added); *id.* at 8:18-26 ("FIG. 3C is a schematic representation of a user interface graphic 'slider' 55, available to users of computers 30 . . . . Slider 55, which is preferably displayed on the screens of computers 30, includes a bar 56 and a movable indicator 58. The symbols J, J+1, J+2, . . . N in the figure are the indices of the slices of stream 40 that are stored on server 36, wherein N is the index of the most recent slice, and J is the index of the earliest stored slice."); *id.* at 10:27-48 ("The operation of client is controlled by a Java applet . . . . The client first reads index file 50 (FIG. 3B), and graphic 56 (FIG. 3C) is displayed by the client, so that a user can decide and indicate at which slice of data stream 40 to begin downloading. Responsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc.") Figure 3C below shows the graphical slider:

FIG. 3C



Id. at Fig. 3C.

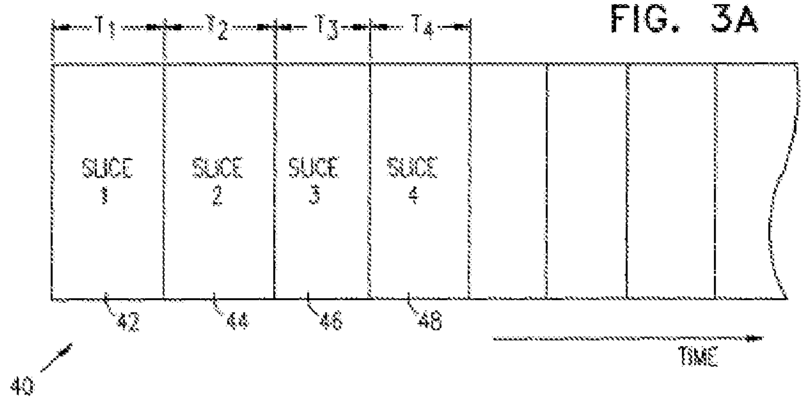
Accordingly, Carmel teaches claim 1[e].

(f) ***1[f] - instructions to cause the media player to play media data elements sequentially from the player buffer; and***

Carmel teaches the media player to play the media data elements sequentially from the player buffer.

Carmel downloads sequential slices from an index file by serial number. *See* claim 1[a] and [c]. The sequential slices are shown in Figure 3A below.

FIG. 3A



Carmel (Ex. 3), at Fig. 3A; *id.* at 2:1-21, 7:22-26.

Carmel teaches that the browser and/or software (Java applet) will decode and play the sequential slices such as in Figure 3A. *Id.* at 10:45-50 ("Responsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc. In the case of a multimedia stream, client 30 reconstructs and outputs the multimedia data for the appreciation of a user."); *id.* at 10:25-31 ("FIG. 6A is a flow chart illustrating the operation of clients 30 in downloading and playing back data stream 40 (FIG. 3A) transmitted by computer 34, in accordance with a preferred embodiment of the present invention. The operation of client is controlled by a Java applet . . . and includes facilities for carrying out the steps shown in FIG. 6A

. . . ."); *id.* at 7:4-17 ("Clients 30 connect to server 36 and receive the multimedia sequence, substantially in real time. Clients 30 preferably download the sequence using the Hypertext Transfer Protocol (HTTP), although other Internet protocols may also be used, such as UDP or RTP . . . . because HTTP is supported by substantially all modern Web browsers, clients 30 will typically need only add a Java applet or plug-in to their existing Web browsers, as described further hereinbelow, in order to receive and play back the broadcast."); *id.* at 2:4-21 ("The data stream is divided into a sequence of segments or slices of the data . . . . Each slice is preferably assigned a respective slice index. . . . . The clients use the slice indices of the frames to maintain proper synchronization of the playback. The division of the data stream into slices and the inclusion of the slice indices in the data stream to be used by the clients in maintaining synchronization allows the broadcast to go on substantially in real time without the use of special-purpose hardware."); *id.* at 11:7-8 ("The slices are received, decoded and output by the client.") This downloading and playing of sequential slices are shown in Figures 6A and 6B. Thus, Carmel teaches claim 1[f].

(g) ***1[g] - instructions to cause the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.***

As discussed, Carmel teaches that the media player requests a predetermined number of slices (*e.g.*, one, two, or five slices) by serial number and repeats those requests. *See* claim 1[c]; Carmel (Ex. 3), at 3:5-9, 7:59-8:5, 8:18-31, 8:56-62, 10:25-50, 10:64-11:8, claim 10, Figs. 3C-3D.

Carmel repeats these requests to maintain a specified number of slices given the player buffer "keep[s] up" with the transmission of slices for uninterrupted playback. Carmel (Ex. 3), at 7:36-40 ("clients 30 similarly monitor the time codes as the file is received, in order to ensure that

the transmission or reception is 'keeping up' with the input of the data to the computer."), 2:15-21. For instance, Carmel teaches downloading specified slices ("42, 44, and 46, etc.,") in the player buffer to ensure this continuous playback. *Id.* at 10:36-54 ("Typically, client 30 opens one or two HTTP links, over which files 42, 44, 46, etc., are downloaded in successive alternation . . . . Time stamps in the data stream are used to synchronize the data, so that the multimedia sequence is played back just as it was input at computer 34, preferably with only a minimal necessary transmission and decoding delay.")

Separately, Carmel teaches to maintain specified slices in the buffer by downloading the appropriate quality slices based on network bandwidth. *Id.* at 9:6-8 ("Each of clients 30 chooses or is assigned the quality level appropriate to the bandwidth of its link on network 28 to server 36."). For example, if the transmission rate is low, then the media player will select a lower quality level slice to maintain the specified number of slices. *Id.* at 11:11-15 ("if the rate is low, such that time stamps 59 indicate that the slices need to be played as fast as or faster than they are being received, the client will preferably select a lower quality level if one is available."); 11:15-22.

And this maintaining of specified slices continues for the duration of the broadcast. *Id.* at 12:46-47 ("This process continues in alternation for all of the slices of stream 40, until transmission is completed."); *id.* at 2:24-25 ("the segments or slices may all be contained in a single indexed file, which is streamed to the client . . . .").

Accordingly, Carmel teaches claim 1[g]. Further, Carmel especially teaches claim 1[g] if WAG's proposed plain and ordinary meaning construction of the predetermined limitation is adopted. As discussed previously, WAG broadly contends that any specified data in the client buffer at any point (whether before the client connects to the server or after the client connects to the server) would cover this claim limitation. *See supra* 18-19 (WAG contending predetermined

is a "specified number of media data elements" that "could be determined after connection to the network and after connection to the media source" and are not constant or the "same number").

Carmel downloads and buffers specified slices by "keeping up" with the transmission and separately downloading certain quality slices. Thus, Carmel especially anticipates under WAG's broad plain and ordinary meaning construction given the client buffers specified data.

To the extent it is argued that Carmel does not teach claim 1[g], it is taught by Hill. As discussed above, a POSITA would have been motivated to combine Carmel and Hill for claim 1[g]. Hill teaches to repeat requests for frames by serial number to maintain a "request threshold" for the duration of the program. *See supra* 59-61. Accordingly, Carmel and Hill teach claim 1[g].

3. **Claim 4**

Claim 4 states, "4. The media player of claim 1, wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player."

Carmel teaches the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player.

As described in claim 1[c] and 1[g], Carmel repeatedly requests a predetermined number of slices. The PTAB previously determined that Carmel teaches receiving the slices at a rate more rapid than the rate at which the media data elements are to be played out by the media player in two ways. *See supra* 43-46; WebPower Final Written Decision on Remand (Ex. 12), at 16-23. Accordingly, Carmel teaches claim 4.

G. SNQP7: Bloch anticipates or Bloch and Carmel, Glaser, or TCP renders obvious claims 1 and 4

1. Motivation to Combine

A POSITA would have found it obvious to combine the teachings of Carmel and Bloch because the combination is merely a matter of applying a known technique to a known device for improvement to yield predictable results, including for claim 4.

Bloch and Carmel disclose similar client-pull systems, including requesting data by serial number. *See supra* 12-13; claim 1[c] below; Bloch (Ex. 8), at Abstract, 5:40-42, 5:42-46, 4:57-61, 6:63-7:3, 9:15-22; Carmel (Ex. 3), at 7:59-8:5, 10:25-48, Figs. 3A, 6A, 7:18-35. Moreover, both maintain a record of the last serial number received. *Id.* A POSITA under *KSR* (*see supra* 20-21) at the time of the purported invention was made would have been motivated to combine the teachings of Bloch with those of Carmel given the similarity of the Carmel and Bloch systems and a POSITA would have been motivated to review other client-pull systems to further efficiency and enhance Bloch's functionality. For instance, a POSITA would have been motivated to combine Carmel's sending faster than playback rate to further enhance Bloch's real-time or smooth playback. Bloch (Ex. 8), at Abstract, 8:8-13, 9:51-58.

A POSITA would have been further motivated to implement such function and would have had a reasonable expectation of success in doing so considering Bloch's existing functionality. Bloch already discloses receiving predetermined frame requests at a high rate and buffering them. *Id.* at Abstract, 2:4-10, 5:42-46, 5:52-59, 6:63-7:3, 7:65-7:67, 8:5-14, 9:51-58. Further, Bloch has existing server structure, including an I/O engine and data storage. *Id.* at 6:36-43. As such, the predetermined frames received in Bloch's player could be received faster than playback and

properly buffered, resulting in further enhanced real-time or smooth playback. *Id.* Thus, Bloch could implement Carmel's sending data faster than the playback rate seamlessly.

Also, a POSITA would have been motivated to combine the references given the patentee's admissions that the TCP protocol was well-known for proper data delivery from server to client and that media players, including client buffers, were well-known for playback. *See supra* 2-3; *Unwired Planet*, 841 F.3d at 1003; *McCoy*, 850 F. App'x at 789. A POSITA would have been motivated to improve on this known knowledge, *e.g.*, transmitting data and storing such data in the client's buffer, by combining Carmel's transmitting faster than the playback rate to further enhance Bloch's efficiency in buffering and playing streaming data. *Advanced Micro Devices*, 848 F.2d at 1570; *WesternGeco*, 889 F.3d at 1308. Accordingly, a POSITA would have been motivated to combine Bloch and Carmel.

Additionally, a POSITA would have been motivated to combine Bloch and Glaser. Bloch and Glaser both teach video distribution systems over the Internet. *See supra* 13, 15; claim 1[a]-[c] below. Glaser transmits multimedia data faster than the playback rate to maintain a client buffer threshold. Glaser (Ex. 9), at 3:15-28, 3:31-36, 13:36-44, 16:2-10, 21:57-63, 22:41-59. A POSITA under *KSR* (*see supra* 20-21) would have been motivated to implement Glaser's sending data faster than the playback rate in Bloch to enhance Bloch's functionality, including smooth playback. *Id.* at Abstract, 2:4-10, 8:5-14, 9:51-58. Also, Glaser is cited on the front page of the '011 Patent, so a POSITA would have been motivated to review and apply it to other prior art, including Bloch.

A POSITA would have been further motivated to implement such function and would have had a reasonable expectation of success in doing so considering Bloch's existing server and client functionality discussed above. For instance, Bloch teaches sending data at a high rate and

buffering frames. Thus, Bloch could buffer the data sent by Glaser faster than the playback rate. Accordingly, a POSITA would have been motivated to combine Bloch and Glaser.

2. **Claim 1**

(a) ***1[a] - A media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number, comprising***

Bloch teaches a media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number.

Bloch teaches a media player requesting and receiving frames from a server over an Internet protocol network. Bloch (Ex. 8), at 4:49-4:52 ("The present invention is directed to a system and method for enabling a user to view media data (e.g., movie data, video data, audio data, meta-data, etc.) using a workstation connected to a digital network."); *id.* at 5:16-23 ("A clip is a sequence of frames. The movie data of interest to a user will be composed of one or more clips. Clips may be stored in different locations within the digital network environment 202. For purposes of discussion, assume that the movie data of interest to the user contains N clips, each stored in a server 222 that is coupled to the workstation 204 via data network 215."); *id.* at 6:15-6:20 ("GUI 214 further includes a window for displaying a playlist. A playlist generally includes a list of sequence of media data clips to be played. Typically, each clip has a 'clip name,' a 'host name,' the beginning frame number and the ending frame number."); *id.* at 1:66-2:2 ("Accordingly, the present invention provides a software architecture and control protocol that

allows frame-accurate, random-access, real-time and stream-based media data over a data network.") The architecture of the client, network, and server are shown in Figure 2 below:

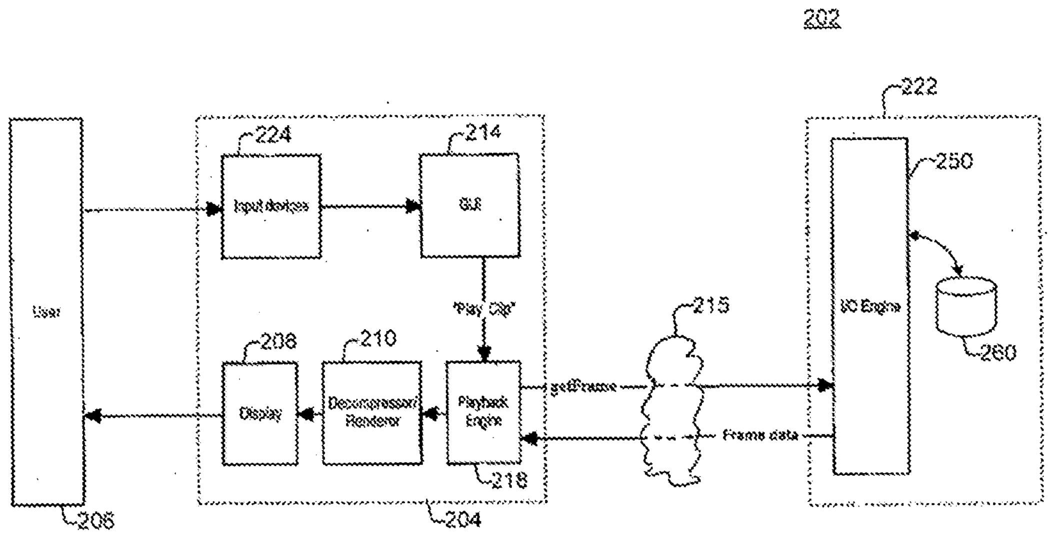


Fig. 2

Id. at Fig. 2; *id.* at Fig. 1; *id.* at 2:52-57 ("The server upon receiving the frame accurate requests retrieves the requsted [sic: requested] frames and transmits them back to the client one frame at a time. An advantage of the present invention is that frame accurate control and playback of server based media data across a data network can be achieved."); *id.* at title ("Frame-accurate transport of media data across a data network").

The client or playback engine repeatedly requests from the server one frame at a time by frame or serial number for playing. *Id.* at Abstract ("The present invention provides control on the client with the implementation of a 'client pull' data transport model. According to the present invention, requests for media data are sent from the client to the server one request at a time. The server, upon receiving the requests, retrieves the corresponding frames of media data and transmits the media data back to the client one frame at a time. In this way, the present invention allows a

user-interface to provide frame accurate controls on the client workstation, while allowing for real-time playback of media data that resides on the server."); *id.* at 5:40-46 ("playback logic is implemented within a playback engine 218 that resides on the client workstation 204. Particularly, the playback engine 218 translates the user control inputs (e.g., 'play clip A') received by GUI 214 into frame accurate requests (e.g., 'Get Frame 1 of clip A'). The frame accurate requests are then sent across data network 215 to server 222."); *id.* at 6:60-7:18 ("For example, GUI 214 may receive control inputs that specify frame 1000 of Clip B to be played. The frame accurate request corresponding to such user control inputs would be 'Get Frame 1000 Clip B.' As another example, GUI 214 may receive control inputs that specify Clip C to be played. In that case, a series of frame accurate requests such as 'Get Frame 1 Clip C,' 'Get Frame 2 Clip C,' 'Get Frame 3 Clip C,' etc. . . . It is important to note that each frame accurate request corresponds to a single frame only. The data structure of an exemplary frame accurate request generated by the playback engine 218 . . . is illustrated below in Table 1. TABLE 1 struct ServerGetFrameReq { int reqType; void* clipId; int frame; bool hasVideo; bool hasAudio; }"); *id.* at 9:51-53 ("At step 616, the client workstation 204 a receives frames of the second clip B, and begins rendering clip B immediately after the last frame of the first clip A is rendered."); *id.* at 9:34-35 ("At step 610, as the client workstation 204 a receives and renders the requested frames."); *id.* at Figs. 3, 6.

Accordingly, Bloch teaches claim 1[a]. Further, Bloch especially teaches 1[a] if WAG's proposed plain and ordinary meaning construction of "serial number / identifier" is adopted. As discussed previously, WAG contends that "serial number/identifier" is not limited to "consecutive [numbers] or limited to numerals," and thus can broadly encompass several forms. *See supra* 18. Bloch teaches that the requested frames (or frame numbers) are not limited to a specific form and indeed can include numeric and consecutive frame numbers. Bloch (Ex. 8), at 5:40-5:51, 6:60-

7:18, 9:51-58, Figs. 3, 6. Thus, Bloch especially anticipates under WAG's broad plain and ordinary meaning construction.

(b) ***1[b] - a processor; a memory; a connection to the network; and***

Bloch describes a processor, memory, and connection to the network. Bloch teaches the client computer includes "a central processor 104 coupled with bus 102 for processing information and instructions, a volatile memory 106 (e.g., random access memory RAM) coupled with the bus 102 for storing information and instructions for the central processor 104 and a non-volatile memory 108 (e.g., read only memory ROM) coupled with the bus 102 for storing static information and instructions for the processor 104." Bloch (Ex. 8), at 4:5-12. This client in Bloch also "includes a data storage device 110 ('disk subsystem') such as a magnetic or optical disk and disk drive coupled with the bus 102 for storing information and instructions." *Id.* at 4:13-15.

Bloch further includes "a communication device 118 (e.g., network interface card) coupled to the bus 102 for interfacing with a network." *Id.* at 4:24-26; *id.* at 4:49-61 ("The present invention is directed to a system and method for enabling a user to view media data (e.g., movie data, video data, audio data, meta-data, etc.) using a workstation connected to a digital network. . . . [D]igitized movie data may be accessed from different sources (e.g., network servers, local disks). Particularly, in the present invention, playback engines are implemented within the clients (e.g., workstations) rather than the server. Because the playback engines are implemented within the clients, frame accurate transportation of media data can be achieved."); *id.* at 4:63-5:6 ("A user 206 utilizes the workstation 204 to play movie data. The workstation 204 may be a conventional personal computer (e.g., computer system 101), or dedicated hardware specially suited for audio/video playback. . . . Significantly, the user 206 may issue a frame accurate play command

to indicate a specific frame to be played even though that frame is stored in a remote location across a data network.") Accordingly, Bloch teaches claim 1[b].

(c) ***1[c] - media player software comprising instructions to cause the media player to request from the media source a predetermined number of media data elements;***

Bloch teaches that the player or playback engine requests a predetermined number of frames for a clip or clips by frame number from the server, *e.g.*, "M requests" for "M frames" of a clip or clips. *See* claim 1[a]; Bloch (Ex. 8), at 5:49-51 ("Thus, in order to play a clip containing M frames, playback engine 218 generates M requests and transmits them to server 222."); *id.* at 5:52-59 ("It should be noted that the playback engine 218 is responsible for automatically generating frame accurate requests when the user control inputs specify a clip or clips to be played. For example, if a clip is requested by the user, playback engine 218 automatically generates frame accurate requests for all the frames of the clip, and transmits the requests to appropriate servers in which the frames are stored."); *id.* at 5:42-46 ("playback engine 218 translates the user control inputs (e.g., 'play clip A') received by GUI 214 into frame accurate requests (e.g., 'Get Frame 1 of clip A'). The frame accurate requests are then sent across data network 215 to server 222."); *id.* at 6:65-67 ("[A] series of frame accurate requests such as 'Get Frame 1 Clip C,' 'Get Frame 2 Clip C,' 'Get Frame 3 Clip C,' etc., ..."); *id.* at 9:10-12 ("At step 604, the client workstation 204 a of the present embodiment translates the user control inputs into frame accurate requests for the requested clips.").

Accordingly, Bloch teaches claim 1[c]. Further, Bloch especially teaches claim 1[c] if WAG's construction of "predetermined" discussed with reference to claim 1[g] as any specified data is applied here. *See supra* 18-19. Bloch requests a predetermined number of frames for a clip

or clips. *See* above. Thus, Bloch especially anticipates under WAG's broad plain and ordinary meaning construction given specified data is requested.

(d) ***1[d] - instructions to cause the media player to receive media data elements sent to the media player by the media source and store the media data elements in the memory;***

Bloch teaches that the media player receives and stores elements in memory. As described in claim 1[b]-[c], Bloch teaches a computer with playback engine and memory. This is shown in Figure 3, 308 "Server retrieves the requested frame and sends the requested frame to client" and 310 "Playback engine sends the frame of data to decompressor/renderer one frame at a time":

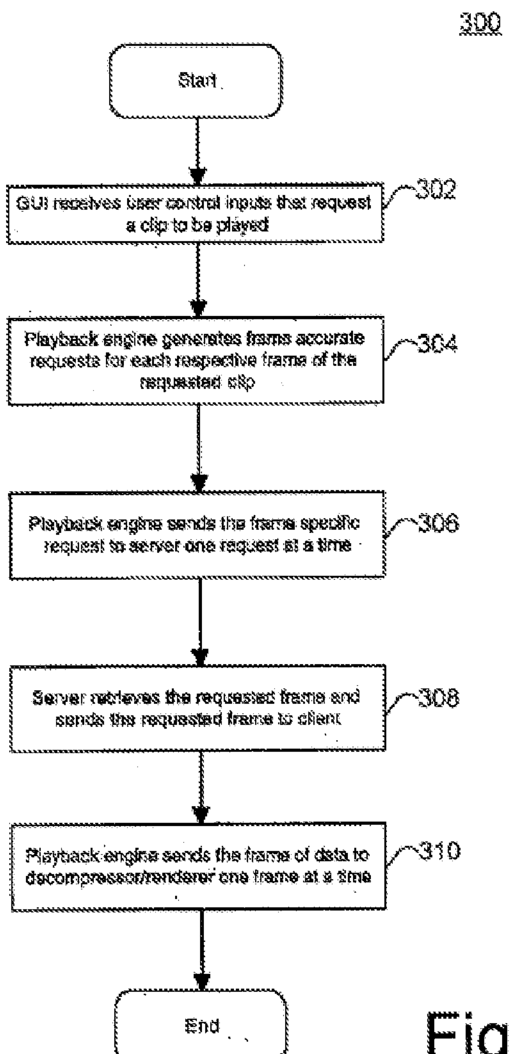


Fig. 3

Bloch (Ex. 8), at Fig. 3; *see also id.* at 8:9-14 ("[T]he playback engine 218 may request the frame several seconds ahead of time and store the retrieved frames in a buffer to ensure smooth playback of the movie data. Thus, in the present embodiment, the playback engine 218 also schedules the time the requested frame is to be played."); *id.* at 9:51-58 ("At step 616, the client workstation 204 a receives frames of the second clip B, and begins rendering clip B immediately after the last frame of the first clip A is rendered. Significantly, in the present embodiment, a constant interval is kept between the frames in the first clip A and between the last frame of clip A and the first frame of the second clip B. In this way, a smooth and seamless transition from the first clip A to the second clip B is achieved."); *id.* at Abstract ("requests for media data are sent from the client to the server one request at a time. . . . In this way, the present invention allows a user-interface to provide frame accurate controls on the client workstation, while allowing for real-time playback of media data that resides on the server.") Accordingly, Bloch teaches claim limitation 1[d].

(e) ***1[e] - instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer;***

Bloch teaches a player buffer manager for managing a player buffer to maintain a record of the serial number of the last element that has been received and stored in the player buffer.

As discussed above, Bloch teaches that the player requests frames by frame or serial number for a clip or clips, *e.g.*, "Get Frame 1." *See* claim 1[a] and 1[c]; *see also* Bloch (Ex. 8), at 5:42-46 ("playback engine 218 translates the user control inputs (e.g., 'play clip A') received by GUI 214 into frame accurate requests (e.g., 'Get Frame 1 of clip A'). The frame accurate requests are then sent across data network 215 to server 222."); *id.* at 5:49-51 ("Thus, in order to play a clip containing M frames, playback engine 218 generates M requests and transmits them to server

222."); *id.* at 6:65-67 ("[A] series of frame accurate requests such as 'Get Frame 1 Clip C,' 'Get Frame 2 Clip C,' 'Get Frame 3 Clip C,' etc., ..."); *id.* at 7:19-22 ("[T]he frame accurate request includes information relevant to the identification of the clip (e.g., clipId), the frame number of the clip and the type of the request."), at 7:4-18, 9:15-22; *id.* at Fig. 2 (frame-numbered requests).

Bloch teaches a player buffer manager, playback engine, to manage the buffering of the received frames associated with frame numbers and transmit such frames to the renderer for display. *See* claim 1[a]-[c]; Bloch (Ex. 8), at 6:57-60 ("the playback engine 218 determines the frames of the clip(s) to be retrieved and determines where the frames are stored based on the user inputs."); *id.* at 8:10-14 ("the playback engine 218 may request the frame several seconds ahead of time and store the retrieved frames in a buffer to ensure smooth playback of the movie data. Thus, in the present embodiment, the playback engine 218 also schedules the time the requested frame is to be played."); *id.* at 7:65-8:4 ("Because the playback engine 218 may retrieve frames at a high rate, the playback engine 218 also schedules when the frames are sent to the decompressor/renderor 210. Playback engine 218 also synchronizes the audio video information of each frame before sending the frame to the decompressor/renderor 210. When the entire clip has been played, the process ends."); *id.* at 7:25-29 ("playback logic is implemented within playback engine 218. That is, playback engine 218 determines the specific frame number of a clip to be played and generates a request for that particular frame based on the user control inputs received by GUI 214."); *id.* at 2:29-31 ("Significantly, the playback logic of the present invention is implemented in software processes residing on the client side.")

The playback engine stores or records such frames with frame numbers, *e.g.*, frame 1, frame 2, including the last frame number received. *Id.* at 7:36-39 ("At step 308, the I/O engine 250 retrieves the requested frames of movie data from the data storage 260, and sends the requested

frames across data network 215 to the playback engine 218 of the workstation 204, one frame at a time."); *id.* at 7:61-65 ("At step 310, as the playback engine 218 receives a frame ready signal and the frame of media data associated therewith, playback engine 218 sends the frame of data to the decompressor/renderor 210 for decompression and rendering. . . ."); *id.* at Abstract ("The server, upon receiving the requests, retrieves the corresponding frames of media data and transmits the media data back to the client one frame at a time."); *id.* at 7:43-60, Figs. 3, 6.

Bloch even provides an example of buffering or recording the frame numbers for a first clip (Clip A) to ensure the frames for the next clip (Clip B) are not requested until all the frames from first clip have been received. In other words, Bloch maintains a record of the last frame number received to know when Clip A ends and when to start requesting frame numbers for Clip B. *Id.* at 9:39-46 ("At step 612, when the all the requested frames of the first clip A have been received, client workstation 204 a begins to send requests for the second clip B to server 222 b. It should be noted that, in the present embodiment, there should be sufficient buffer within client workstation 204 a for storing the **frames such that the last frame of the first clip A is not rendered before the request for the first frame of of [sic] the second clip B is sent**.") (emphasis added). Thus, Bloch's media player maintains a record of the identifier of the last data element that has been received by buffering or recording each requested frame number.

Additionally, and separately, Bloch teaches maintaining a record of the last frame number received by virtue of a graphical user interface which records and displays the frame numbers last requested. *Id.* at 5:63-6:7 ("As shown, GUI 214 a display window 750 for displaying the frames and **a frame number window 760 for displaying the frame number of the frame that is currenlty** [sic: currently] **played**. . . . Significantly, GUI 214 includes a slider 740 that indicates the **position of the currently played frame** within the clip. GUI 214 allows **users to select a**

particular frame to be displayed within display window 750 by moving slider 740.") (emphasis added). The graphical user interface with the frame numbers is shown below:

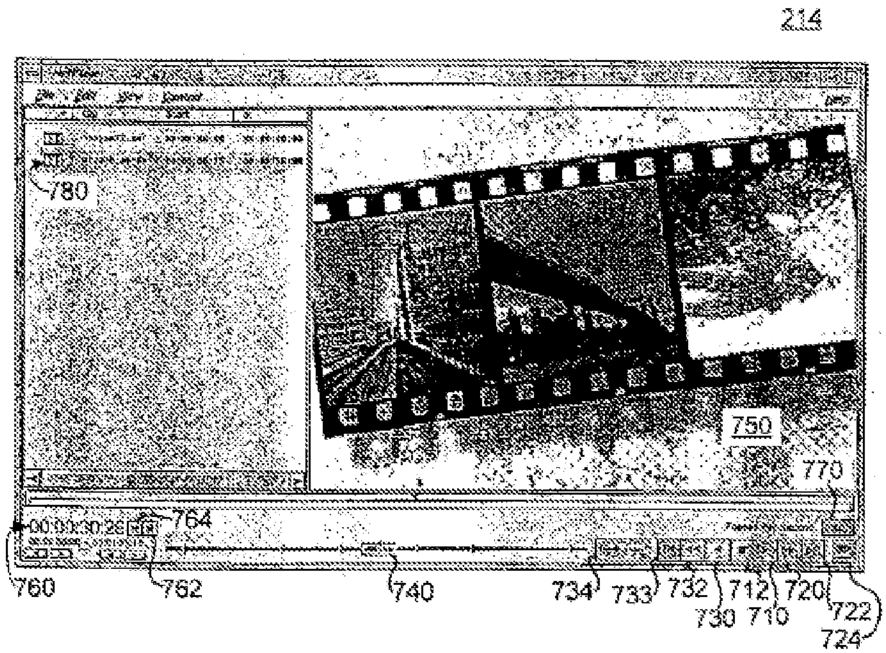


Fig. 7

Id. at Fig. 7; *id.* at 3:18-20. Accordingly, Bloch teaches claim 1[e].

(f) ***1[f] - instructions to cause the media player to play media data elements sequentially from the player buffer; and***

Bloch teaches that the media player plays elements sequentially from the player buffer. Specifically, the playback engine manages the buffer and sends the sequential frames for playing or rendering. *See* claim 1[c]-[e] (sequential frames sent to the renderer for display); Bloch (Ex. 8), at 7:61-8:4 ("At step 310, as the playback engine 218 receives a frame ready signal and the frame of media data associated therewith, playback engine 218 sends the frame of data to the decompressor/renderor 210 for decompression and rendering. Because the playback engine 218 may retrieve frames at a high rate, the playback engine 218 also schedules when the frames are

sent to the decompressor/renderor 210. Playback engine 218 also synchronizes the audio video information of each frame before sending the frame to the decompressor/renderor 210. When the entire clip has been played, the process ends."); *id.* at 9:51-53 ("At step 616, the client workstation 204 a receives frames of the second clip B, and begins rendering clip B immediately after the last frame of the first clip A is rendered."); *id.* at 9:34-34 ("At step 610, as the client workstation 204 a receives and renders the requested frames."); *id.* at Abstract, Fig. 6. Figure 3 shows the sequential playing, 308 "Server retrieves the requested frame and sends the requested frame to client" and 310 "Playback engine sends the frame of data to decompressor/renderer one frame at a time":

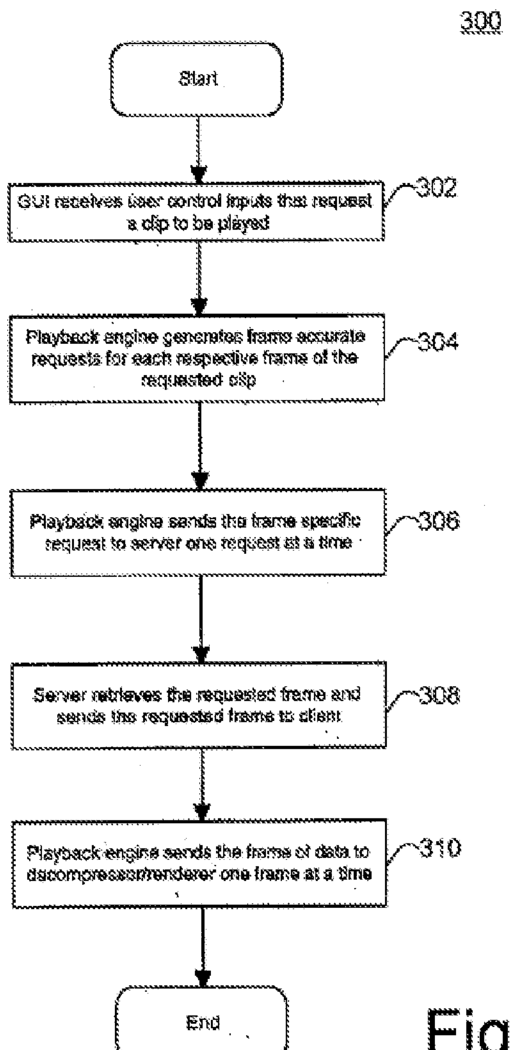


Fig. 3

Bloch (Ex. 8), at Fig. 3. Accordingly, Bloch teaches claim 1[f].

(g) ***1[g] - instructions to cause the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.***

Bloch teaches that the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.

As discussed in claim 1[a] and [c], Bloch teaches that the playback engine requests frames for a clip or clips from the server by frame number or serial number.

Bloch teaches that those requests are repeatedly made to maintain a predetermined number of frames in the player buffer because Bloch requests the frames of a clip or clips and buffers all such frames (*e.g.*, clip A). *See* claim 1[c]; Bloch (Ex. 8), at 9:39-46 ("At step 612, when the **all the requested frames of the first clip A have been received**, client workstation 204 a begins to send requests for the second clip B to server 222 b. It should be noted that, in the present embodiment, there **should be sufficient buffer within client workstation 204 a for storing the frames such that the last frame of the first clip A** is not rendered before the request for the first frame of of [sic] the **second clip B is sent**.") (emphasis added); *id.* at 5:49-51 ("Thus, in order to play a clip containing M frames, playback engine 218 generates M requests and transmits them to server 222."); *id.* at 9:51-53 ("At step 616, the client workstation 204 a receives frames of the second clip B, and begins rendering clip B immediately after the last frame of the first clip A is

rendered."); *id.* at claim 1 ("… said client translating said playback control signal into a plurality of requests corresponding to respective individual frames of said clip ... said server retrieving said respective individual frames ... transmitting said respective individual frames to said client one frame at a time."). Figure 6 shows storing all the frames for clip A, specifically 610 "Client receives and renders the requested frames of Clip A" and 612 "When all requested frames of Clip A have been received, client sends requests for frames of Clip B to second server":

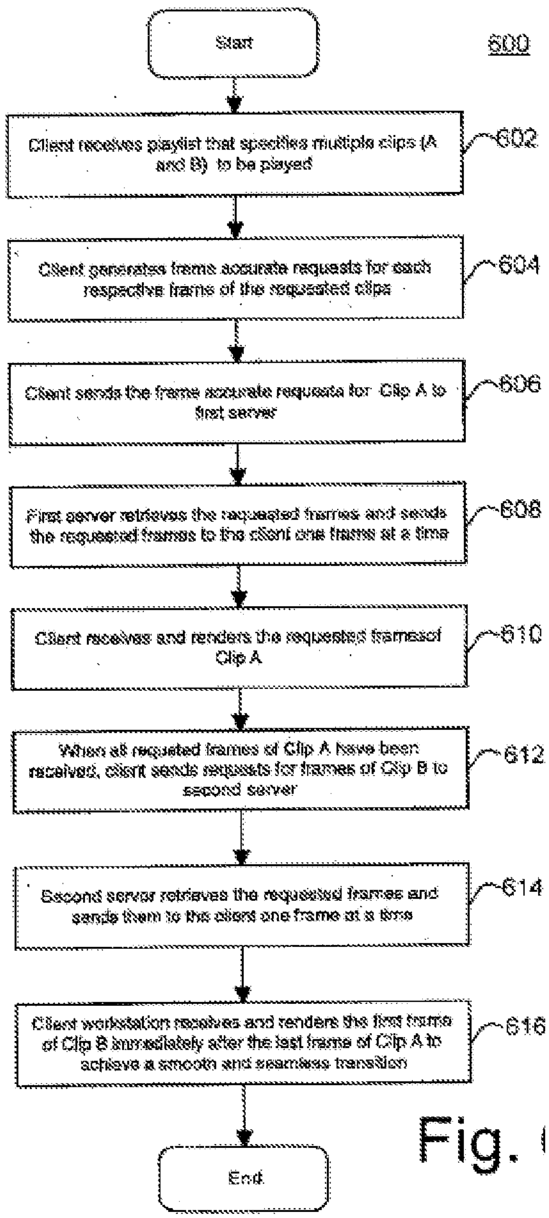


Fig. 6

Id. at Fig. 6; *id.* at Fig. 3.

The frames are requested by the media player until the end of the program or clip(s). *Id.* at 7:61-8:4 ("At step 310, as the playback engine 218 receives a frame ready signal and the frame of media data associated therewith, playback engine 218 sends the frame of data to the decompressor/renderor 210 for decompression and rendering. . . . Playback engine 218 also synchronizes the audio video information of each frame before sending the frame to the decompressor/renderor 210. **When the entire clip has been played, the process ends.**") (emphasis added); *id.* at 5:16-17 ("A clip is a sequence of frames. The movie data of interest to a user will be composed of one or more clips."); *id.* at Fig. 6 above (discussing requesting frames for Clip A, Clip B, and then ending); *id.* at 9:51-53 ("At step 616, the client workstation 204 a receives frames of the second clip B, and begins rendering clip B immediately after the last frame of the first clip A is rendered.").

Accordingly, Bloch teaches claim 1[g]. Further, Bloch especially teaches claim 1[g] if WAG's proposed plain and ordinary meaning construction of the predetermined limitation is adopted. As discussed previously, WAG broadly contends that any specified data in the client buffer at any point (whether before the client connects to the server or after the client connects to the server) would cover this claim limitation. *See supra* 18-19 (WAG contending predetermined is a "specified number of media data elements" that "could be determined after connection to the network and after connection to the media source" and are not constant or the "same number").

Bloch requests and stores all the frames of a clip or clips. *See* above. Thus, Bloch especially anticipates under WAG's broad plain and ordinary meaning construction given the client buffers specified data.

3. **Claim 4**

Claim 4 states, "4. The media player of claim 1, wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player."

Bloch teaches requesting or receiving predetermined frames of the clip(s) at a high rate and buffering such high rate frames. Bloch (Ex. 8), at 7:65-67 ("Because the playback engine 218 may retrieve frames at a high rate, the playback engine 218 also schedules when the frames are sent to the decompressor/renderor 210."); *id.* at 8:10-14 ("the playback engine 218 may request the frame several seconds ahead of time and store the retrieved frames in a buffer to ensure smooth playback of the movie data. Thus, in the present embodiment, the playback engine 218 also schedules the time the requested frame is to be played.") Accordingly, Bloch teaches claim 4.

To the extent it is argued that Bloch does not teach claim 4, it is taught by Carmel. As discussed above, a POSITA would have been motivated to combine Bloch and Carmel for claim 4. Bloch already describes the client receiving a predetermined number of frames. *See* claim 1[g]. Carmel teaches that the media player receives the media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player. *See supra* 43-46. Accordingly, Bloch and Carmel teach claim 4.

Additionally, Bloch and Glaser or TCP teach claim 4. A POSITA would have been motivated to combine Bloch and Glaser or TCP for claim 4. Bloch already describes the client receiving a predetermined number of frames. *See* claim 1[g]. Glaser or TCP teaches the client receiving data at a rate faster than the playback rate. *See supra* 47. Accordingly, Bloch and Glaser

or TCP teach claim 4.

H. SNQP8: Kliger anticipates claims 1 and 4

1. Claim 1

(a) *1[a] - A media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number, comprising*

Kliger teaches a media player for receiving an audio or video program, the program comprising media data elements, from a media source over the Internet, and playing the program for a user, wherein each of the media data elements is associated with a serial number.

Kliger describes a media player requesting and receiving a sequence of objects (*e.g.*, video) over the Internet as well as playing them. Kliger (Ex. 5), at Abstract ("[A] data stream is formed of a sequence of requested objects. The defined order of the sequence of objects is determined from a client request for data."); *id.* at 6:23-27 ("the present invention is employed on what has become known as the Internet (an international computer network linking an estimated 35 million people to approximately 4.8 million host computers or information sources) ."); *id.* at 9:24-9:29 ("objects may include text, graphics, audio, video, and other types of digitized information. Furthermore, objects may be complete data files or only portions of such files. In addition, the objects themselves may be classes that consist of a number of objects grouped together."); *id.* at 11:34-12:5 ("The streaming server 17 then assembles the stream block 310a as illustrated, which includes all of the objects that the client 13a has requested. In the illustrated example this includes objects (O4a, O7, O2, O12, O3) . After assembly of the stream block 310a the streaming server 17 then provides the stream block 310a as a single object stream 300 to the client 13a."); *id.* at 7:10-12 ("client 13a is a computer that executes an application (or other user interaction) for which

certain data needs to be made present."); *id.* at Fig. 2 ("Xfer Object Stream" is sent to the "116 Receive Data Process" and then delivered to "100 Application/User Interaction"); *id.* at 9:6-9 ("On the receiving end, the client 13a receives the object stream in state 116. More accurately, the data of the object is then delivered to the requesting application of the client for further processing and use.") Figure 1 below is a "computer network employing the present invention."

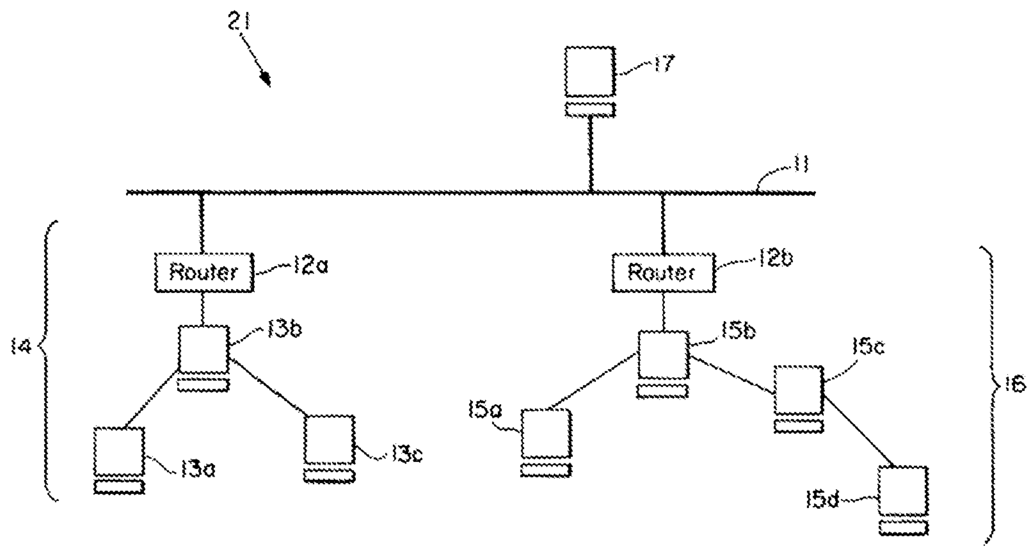


FIG. 1

Id. at Fig. 1, 6:7-7:9.

Kliger further teaches that the client application requests from the server an object map listing all objects (*e.g.*, video, audio) by serial number the client application seeks. *Id.* at 7:13-22 ("the client receives an object stream request from the application that includes a global identification of an object map which indicates certain objects existing in the network 21 that the client application program seeks. In particular, the object stream request includes a list, or linear sequence of objects identified a global object identification number. The client 13a transmits the initial request to the server 17, which enters state 200 to transfer back a global identification of the object map listing subject objects."); *id.* at 7:28-8:2 ("If the object map is not found locally, then

the client requests the object map from the server in state 106. . . . the server . . . (i) initialize a log of object identifications for this client . . . and (iii) transmit the object map identified by the client 13a."); *id.* at Fig. 2 (106 "Request Object Map").

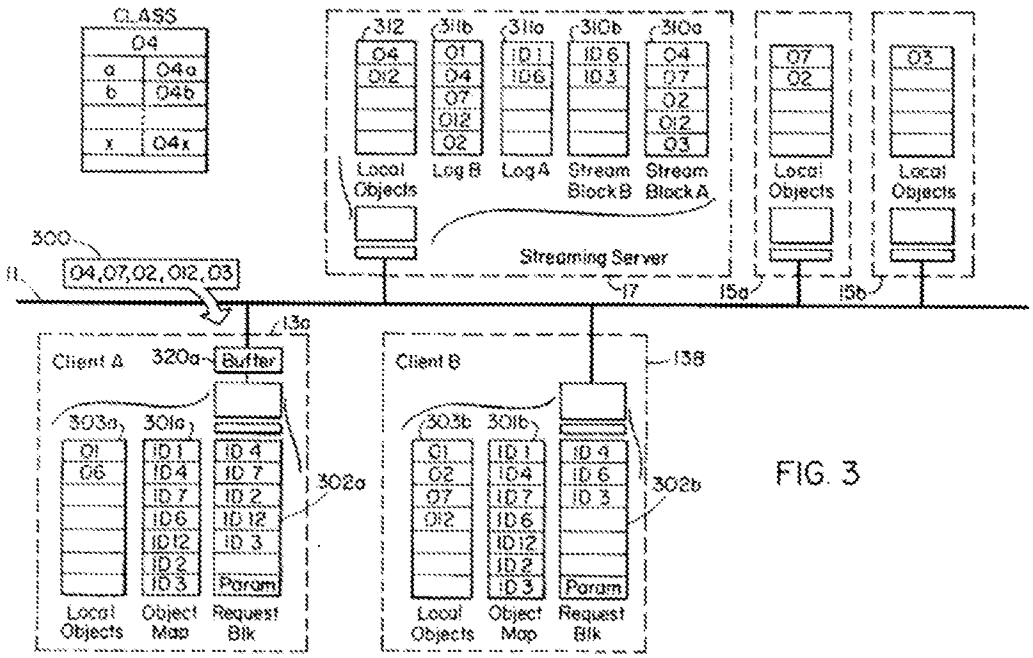
Thereafter, the client requests by identification or serial number the objects from the object map. *Id.* at 8:16-20 ("Upon receipt of this request (i.e., a block of object indentifications [sic: identifications]), the server 17 then initiates the assembly of a data stream and compiles the stream of objects based on (i) the sequence of requested objects and/or (ii) quality of the objects."); *id.* at 10:25-28 ("client 13a already has local objects (O1, O6) available locally. After compiling the request block 302, the request block 302 is sent to the streaming server 17."); *id.* at 8:13-15 ("[T]he client transmits a request to the server for the block of objects as a list of object identifications."); *id.* at 11:34-12:5 ("The streaming server 17 then assembles the stream block 310a as illustrated, which includes all of the objects that the client 13a has requested. In the illustrated example this includes objects (O4a, O7, O2, O12, O3) . After assembly of the stream block 310a the streaming server 17 then provides the stream block 310a as a single object stream 300 to the client 13a."); *id.* at 12:28-30 ("when clients 13a and 13b request that object O4 be provided to them, client 13a may actually receive object O4a and client 13b may receive object O4b . . . .")

Accordingly, Kliger teaches claim 1[a]. Further, Kliger especially teaches 1[a] if WAG's proposed plain and ordinary meaning construction of "serial number / identifier" is adopted. As discussed previously, WAG contends that "serial number/identifier" is not limited to "consecutive [numbers] or limited to numerals," and thus can broadly encompass several forms. *See supra* 18. Kliger's object identifications are not limited to any particular form and can even include a number in consecutive order. Kliger (Ex. 5), at 8:3-15, 8:16-8:25, 10:12-33, 11:34-12:5, 12:17-32, Fig. 3. Thus, Kliger especially anticipates under WAG's broad plain and ordinary meaning construction.

(b) ***1[b] - a processor; a memory; a connection to the network; and***

Kliger teaches a processor and connection to the network. Kliger (Ex. 5), at 6:14-6:23 ("Illustrated in Fig. 1 is a block diagram of a general computer network 21. A plurality of computers 13, 15, 17 are coupled across a communication channel 11 for communication amongst each other. Various subsets of the computers may themselves form a local area network (LAN) or other local network 13, 15. Each of the various local networks are coupled through a respective router 12a, 12b to channel 11. This enables communication from one local network to another across the channel 11 to form what is known as an 'internet'."); *id.* at 6:28-7:2 ("In the preferred embodiment the computers 13, 15, 17 or digital processors employing the present invention are of the PC or mini computer type, or the like, having processing capabilities of the Intel XX386 processing chip or better."); 7:10-12 ("client 13a is a computer that executes an application (or other user interaction) for which certain data needs to be made present."); *id.* at 6:23-27 ("the present invention is employed on what has become known as the Internet (an international computer network linking an estimated 35 million people to approximately 4.8 million host computers or information sources) ."). The client computers are shown in Figure 1 above.

Kliger also teaches memory by virtue of memory or buffers. *Id.* at 7:24-27 ("the client 13a checks multiple local memories such as hardware caches, working memory, CD-Rom's and local network memories for the object map in state 108."); *id.* at 16:2-4 ("'Local' means in various memories including hard drives, cache memories, CD's and other working memories in the local network involving the client 13a."). The client buffering (*e.g.*, 320a "Buffer") or memory is shown in Figure 3 below:



Id. at Fig. 3. Accordingly, Kliger teaches claim 1[b].

(c) ***1[c] - media player software comprising instructions to cause the media player to request from the media source a predetermined number of media data elements;***

Kliger teaches the media player to request from the media source a predetermined number of media data elements.

Kliger requests predetermined objects because it requests a sequence of objects by object ID or serial number from the object map. *See* claim 1[a]; Kliger (Ex. 5), at claim 1 ("in the **requesting processor, forming a request for a desired sequence of objects**; **transmitting** the **formed request** across the communication channel from the requesting processor to the server processor; in the server processor, in response to receipt of the request, assembling and **transmitting, across the communication channel to the requesting processor, a data stream based on the desired sequence of objects**, such that a set of objects in a defined order is transmitted in the data stream from the server processor to the requesting processor.") (emphasis

added); *id.* at 8:3-15 ("The client 13a then analyzes the object map in state 108. . . . For each such object, the client first determines, in state 110, whether the object is in local cache. If not, then the client 13a adds the identification of the object to a request block. . . . When the block is full in state 114, the client transmits a request to the server for the block of objects as a list of object identifications."); *id.* at 8:16-8:28 ("Upon receipt of this request (i.e., a block of object indentifications [sic: identifications]), the server 17 then initiates the assembly of a data stream and compiles the stream of objects . . . . To accomplish this the server constructs blocks of objects to form the data stream, in states 206 through 214. In constructing the blocks, the server 17 maintains a log of objects being transmitted to fulfill the client's request. Specifically, in states 206 and 208, for each object in the block, the server 17 first determines from the log whether the object has previously been transmitted to the particular client 13a.").

This request for a predetermined sequence of objects by object ID is shown in Figure 3 below (client 13a "Request Blk" 302a and client 13b "Request Blk" 302b).

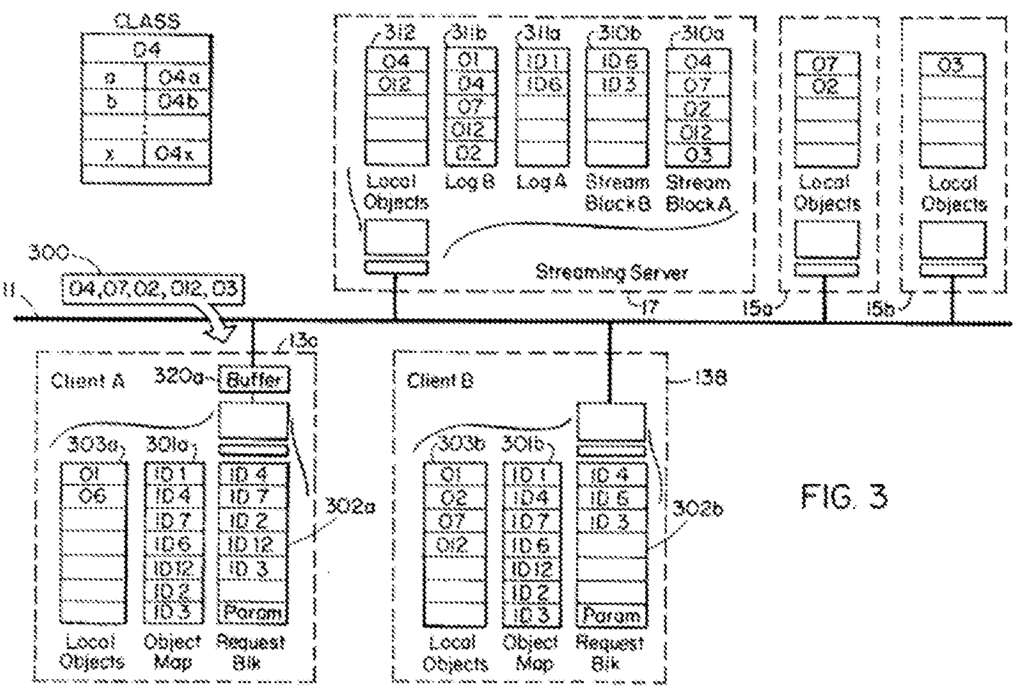


FIG. 3

Id. at Fig. 3. Specifically, client 13a above has a request block ("Request Blk") denoted as 302a of object identifiers (ID4, ID7, ID2, ID12, ID3) and client 13b has a request block denoted as 302b of object identifiers (ID4, ID6, ID3); both requests are for objects not previously requested and stored by the clients from the object map. *Id.* at 8:23-8:28 ("In constructing the blocks, the server 17 maintains a log of objects being transmitted to fulfill the client's request. Specifically, in states 206 and 208, for each object in the block, the server 17 first determines from the log whether the object has previously been transmitted to the particular client 13a."); *id.* at 12:20-23 ("because a different list of object 303b is available to client 13b, the request block 302b created by client 13b will have different object identification numbers (ID4, ID6, ID3) ."); *id.* at 11:34-12-3 ("The streaming server 17 then assembles the stream block 310a as illustrated, which includes all of the objects that the client 13a has requested. In the illustrated example this includes objects (O4a, O7, O2, O12, O3) ."); *see also* Fig. 2 (see steps 108, 112).

Accordingly, Kliger teaches claim 1[c]. Further, Kliger especially teaches claim 1[c] if WAG's construction of "predetermined" discussed with reference to claim 1[g] as any specified data is applied here. *See supra* 18-19. Kliger describes requesting a sequence of objects by object ID or serial number from the object map. *See* above. Thus, Kliger especially anticipates under WAG's broad plain and ordinary meaning construction given specified data is requested.

(d) ***1[d] - instructions to cause the media player to receive media data elements sent to the media player by the media source and store the media data elements in the memory;***

As described in claim 1[a]-[c], Kliger teaches that the media player receives objects sent to the media player by the streaming server and store the objects in the memory. *See also* Kliger (Ex. 5), at 9:6-12 ("On the receiving end, the client 13a receives the object stream in state 116. More accurately, the data of the object is then delivered to the requesting application of the client

for further processing and use. In summary, for each such object request by the client 13a, an object streaming type communication from the server 17 to client 13a is performed."); *id.* at 11:34-12:5 ("The streaming server 17 then assembles the stream block 310a as illustrated, which includes all of the objects that the client 13a has requested. In the illustrated example this includes objects (O4a, O7, O2, O12, O3) . After assembly of the stream block 310a the streaming server 17 then provides the stream block 310a as a single object stream 300 to the client 13a."); *id.* at 12:20-30 ("because a different list of object 303b is available to client 13b, the request block 302b created by client 13b will have different object identification numbers . . . . [A]n object of a particular quality may be targeted to client 13b which is different than as that supplied to client 13a. For example, when clients 13a and 13b request that object O4 be provided to them, client 13a may actually receive object O4a and client 13b may receive object O4b . . . .")

Accordingly, Kliger teaches claim 1[d].

(e) ***1[e] - instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer;***

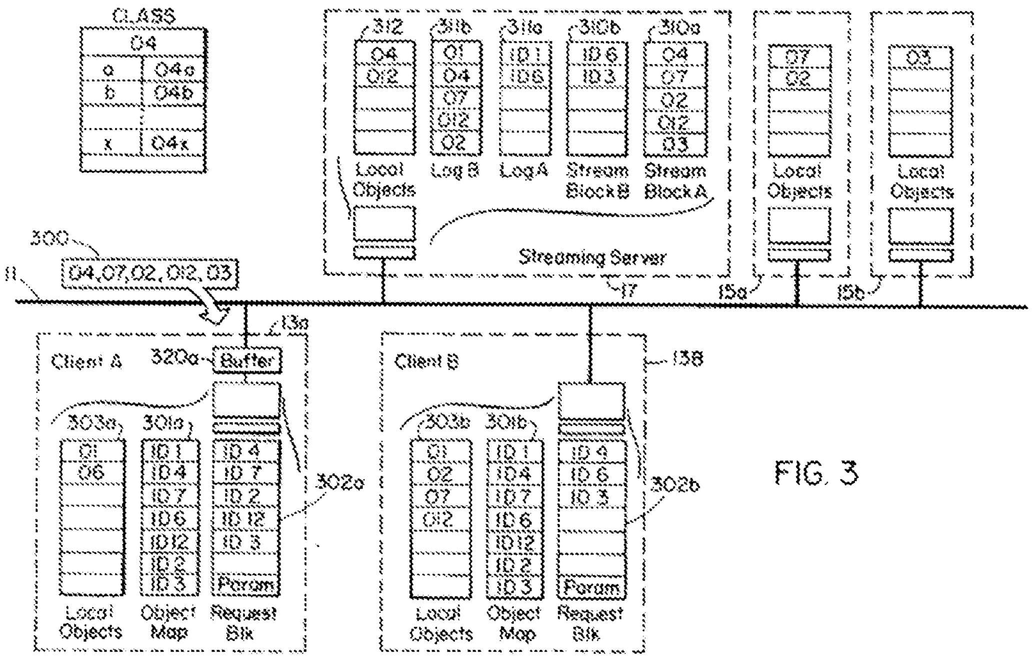
Kliger teaches a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer.

Kliger describes a player buffer manger or client's "application" that manages the buffers. Kliger (Ex. 5), at Fig. 3 (shown below client 13a and client 13b each managing their respective buffer of objects); *id.* at Fig. 2 ("Xfer Object Stream" is sent to the "116 Receive Data Process" and then delivered to "100 Application/User Interaction"); *id.* at 9:6-9 ("On the receiving end, the client 13a receives the object stream in state 116. More accurately, the data of the object is then

delivered to the requesting application of the client for further processing and use."); *id.* at 5:29-31 ("The system may also include buffering, so that the rendering of the set of objects may be delayed until a sufficient amount of data is received at the client."); *id.* at 13:23-29 ("The map tells how fast data is being consumed (used by the client 13a application) . The client then measures the throughput (client receipt) of data in real time and contrasts that with the formulated map. Based on the comparison, the client 13a is able to determine how much of the object stream data 300 should be read by a buffer 320a at a time."); *id.* at 15:1-6 ("The client 13a then calculates the buffer debt from time to time. For a buffer debt greater than 0 the client calculates a maximum wait time . . . which equals the number of seconds of pausing needed to rectify the buffer debt."); *id.* at 7:10-12 ("Referring to Fig. 2, the client 13a is a computer that executes an application (or other user interaction) for which certain data needs to be made present.")

Kliger's client application teaches multiple examples in Figure 3 and related description of maintaining a record of the last data element received. First, Kliger describes that client 13a receives from the server objects denoted as numeral 300 (O4, O7, O2, O12, and O3) and will buffer them denoted as numeral 320a, including the last one received of O4, O7, O2, O12, and O3. *Id.* at 11:34-12:5 ("The streaming server 17 then assembles the stream block 310a as illustrated, which includes all of the objects that the client 13a has requested. In the illustrated example this includes objects (O4a, O7, O2, O12, O3) . After assembly of the stream block 310a the streaming server 17 then provides the stream block 310a as a single object stream 300 to the client 13a."); *id.* at 12:26-30 ("[A]n object of a particular quality may be targeted to client 13b which is different than as that supplied to client 13a. For example, when clients 13a and 13b request that object O4 be provided to them, client 13a may actually receive object O4a and client 13b may receive object O4b . . . .")

This example is shown below in Figure 3:



Id. at Fig. 3.

Second, Kliger teaches in Figure 3 above that client 13a already has knowledge of previously requested objects (O1 and O6) in local storage denoted as numeral 303a and client 13b has knowledge of previously requested objects (O1, O2, O7, O12) denoted as numeral 303b. *Id.* at 10:20-26 ("According to the process already described in connection with Fig. 2, the client 13a first creates a request block 302 taking into account any local objects 303 which it may already have available. . . . client 13a already has local objects (O1, O6) available locally."), Fig. 3; *id.* at 9:14-19 ("the server 17 initializes a log of objects according to object identifications, and then fulfills the request for objects by transmitting the requested objects, in sequence order, which have not previously been transmitted to that client 13a as indicated by the log."). Thus, each client maintains a record of the last object received and buffered in local storage.

Accordingly, Kliger teaches claim 1[e].

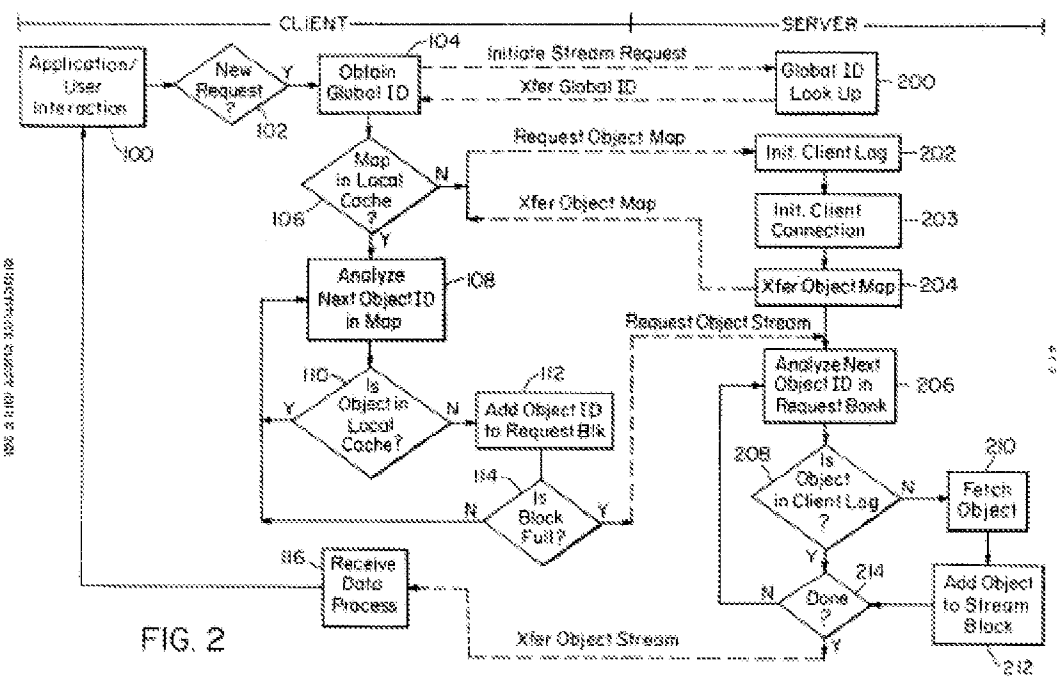
(f) ***1[f] - instructions to cause the media player to play media data elements sequentially from the player buffer; and***

Kliger teaches the media player to play media data elements sequentially from the player buffer.

Kliger describes that the server provides the objects to the client in sequence or defined order. Kliger (Ex. 5), at Abstract ("In a distributed computing environment, a data stream is formed of a sequence of requested objects. The defined order of the sequence of objects is determined from a client request for data."); *id.* at 4:17-20 ("in response to a user or application request for a file, the client issues a request for the file in the form of a sequence of desired objects."); *id.* at 8:16-8:20 ("Upon receipt of this request (i.e., a block of object indentifications [sic: indentifications]), the server 17 then initiates the assembly of a data stream and compiles the stream of objects based on (i) the sequence of requested objects and/or (ii) quality of the objects."); *id.* at 7:10-19 ("Referring to Fig. 2, the client 13a is a computer that executes an application (or other user interaction) for which certain data needs to be made present. . . . [T]he object stream request includes a list, or linear sequence of objects identified a global object identification number.")

As described in claim 1[e], Kliger teaches storing the requested objects in the buffer, which are sequentially played out by the client application. *Id.* at 9:6-12 ("On the receiving end, the client 13a receives the object stream in state 116. More accurately, the data of the object is then delivered to the requesting application of the client for further processing and use. In summary, for each such object request by the client 13a, an object streaming type communication from the server 17 to client 13a is performed."); *id.* at 3:8-11 ("The client may perform a caching or buffering operation prior to actual play back of the media file. This ensures that the media file is

played back to the user of the client computer in a continuous stream."); *id.* at 9:24-27 ("objects may include text, graphics, audio, video, and other types of digitized information. Furthermore, objects may be complete data files or only portions of such files.") This sequential playing is illustrated in Figure 2 below, where the "Xfer Object Stream" is sent to the 116 "Receive Data Process" and then delivered to 100 "Application/User Interaction":



Id. at Fig. 2

Accordingly, Kliger teaches claim 1[f].

(g) ***1[g] - instructions to cause the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.***

Kliger teaches that the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the

requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.

As discussed in claim 1[c], Kliger teaches to transmit a request for a sequence of objects by object ID or serial number from the object map.

Kliger further teaches that the application repeats those requests for objects to maintain a predetermined number of objects by storing those requested or specified objects in the player buffer. *See* claim 1[c]; Kliger (Ex. 5), at claim 1 ("in the requesting processor, forming a **request for a desired sequence of objects**; transmitting the formed request across the communication channel from the requesting processor to the server processor; in the server processor, in response to receipt of the request, assembling and transmitting, across the communication channel to the requesting processor, **a data stream based on the desired sequence of objects, such that a set of objects in a defined order is transmitted in the data stream from the server processor to the requesting processor**.") (emphasis added); *id.* at 8:7-15 ("[T]he client first determines, in state 110, whether the object is in local cache. If not, then the client 13a adds the identification of the object to a request block. . . . . [T]he client transmits a request to the server for the block of objects as a list of object identifications.").

This repeated requesting and buffering of predetermined objects from the object map is shown in Figure 3 below. Specifically, client 13a has previously requested and buffered 303a O1 and O6 while client 13b has previously requested and buffered 303b O1, O2, O7, and O12 from the object map. *Id.* at 10:20-26 ("According to the process already described in connection with Fig. 2, the client 13a first creates a request block 302 taking into account any local objects 303 which it may already have available. . . . client 13a already has local objects (O1, O6) available

locally."); *id.* at 12:20-23 ("because a different list of object 303b is available to client 13b, the request block 302b created by client 13b will have different object identification numbers (ID4, ID6, ID3) .") Each client then sends another request for predetermined objects in the object maps (301a and 301b): client 13a has a request block ("Request Blk") denoted as 302a of object identifiers (ID4, ID7, ID2, ID12, ID3) and client 13b has a request block denoted as 302b of object identifiers (ID4, ID6, ID3).

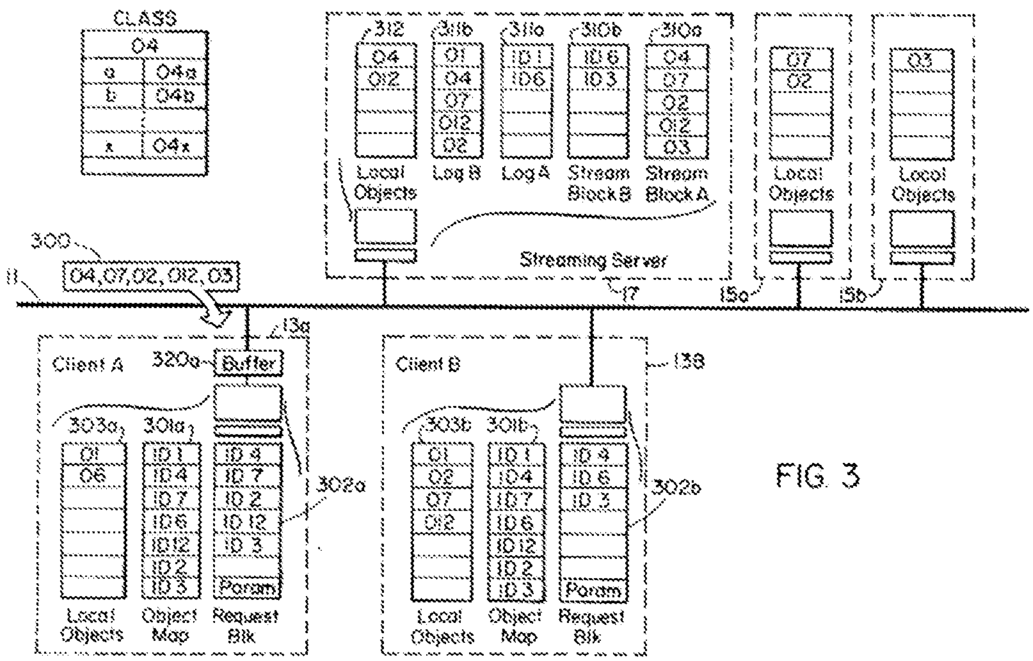


FIG. 3

Id. at Fig. 3. The clients will thereafter buffer these predetermined requested objects. *See* Fig. 3 (shows 300 "O4, O7, O2, O12, O3" being inserted into 320a "Buffer"); *id.* at 11:34-12-5 ("The streaming server 17 then assembles the stream block 310a as illustrated, which includes all of the objects that the client 13a has requested. In the illustrated example this includes objects (O4a, O7, O2, O12, O3) . After assembly of the stream block 310a the streaming server 17 then provides the stream block 310a as a single object stream 300 to the client 13a.").

This continues for the duration of the program given the client (as discussed at *supra* 115) requests all the objects from the object map which represents the data for the client application. *Id.* at 7:13-17 ("the client receives an object stream request from the application that includes a global identification of an object map which indicates certain objects existing in the network 21 that the client application program seeks. In particular, the object stream request includes a list, or linear sequence of objects identified a global object identification number. The client 13a transmits the initial request to the server 17, which enters state 200 to transfer back a global identification of the object map listing subject objects."); *id.* at Fig. 2 ("Xfer Object Stream" is sent to the "116 Receive Data Process" and then delivered to "100 Application/User Interaction"); *id.* at 9:6-9 ("the client 13a receives the object stream in state 116. More accurately, the data of the object is then delivered to the requesting application of the client for further processing and use.")

Accordingly, Kliger teaches claim 1[g]. Further, Kliger especially teaches claim 1[g] if WAG's proposed plain and ordinary meaning construction of the predetermined limitation is adopted. As discussed previously, WAG broadly contends that any specified data in the client buffer at any point (whether before the client connects to the server or after the client connects to the server) would cover this claim limitation. *See supra* 18-19 (WAG contending predetermined is a "specified number of media data elements" that "could be determined after connection to the network and after connection to the media source" and are not constant or the "same number").

Kliger's application repeatedly requests and then buffers the objects requested from the object map. *See* above. Thus, Kliger especially anticipates under WAG's broad plain and ordinary meaning construction given the client buffers specified data.

2. **Claim 4**

Claim 4 states, "4. The media player of claim 1, wherein the instructions for causing the

media player to request from the media source a predetermined number of media data elements further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player."

Kliger teaches that the media player receives the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player.

As described in claim 1[c] and 1[g], Kliger requests and receives a stream of predetermined media objects. Kliger describes receiving such predetermined objects at a sending rate greater than the consumption or playback rate. Kliger (Ex. 5), at 15:10-13 ("[f]or a buffer debt of less than zero, the server 17 transmission of data is ahead of the consumption and thus no pausing during the transmission of the object data stream 300 is warranted."); *id.* at 13:33-14:2 ("client 13a continually **monitors the real data throughput versus data consumption**. The client 13a wants the **delivery-to-consumption ratio to be greater than one so that the throughput (supply) is keeping up with the consumption (demand)**.") (emphasis added); *id.* at 18:21-19:2 ("5. A method as claimed in Claim 1 further including the step of monitoring a rate at which the requesting processor uses requested objects such that rate at which the requesting processor receives requested objects transmitted from the server processor is sufficiently fast to prevent the data stream from lagging behind the requesting processor use of requested objects."). Thus, Kliger teaches receiving the requested predetermined objects at rate more rapid than the rate at which the object are to be played out or consumed.

Additionally, and separately, Kliger teaches receiving the predetermined objects faster than playback rate because the client can request different bandwidth objects based on network

bandwidth (termed bandwidth scalability). This means the client would request each object with a bandwidth less than the available transmission bandwidth and thus receive objects faster than the playback rate. *Id.* at 13:1-8 ("[D]epending on a periodically **calculated data transfer rate**, the **client 13a** or server 17 may choose to **request or transmit a data stream of different objects**. That is, **for a given requested object 04**, a **specific version 04a, 04b...04x may be selected for transfer from the server 17 to the client** 13a as **a function of available bandwidth**. This function is termed '**bandwidth scalability**' of the object stream.") (emphasis added); *id.* at 12:23-32 ("the client parameter(s) provided with the request block by client 13b may very well be different for that provided by client 13a. Thus, an object of a particular quality may be targeted to client 13b which is different than as that supplied to client 13a. For example, when clients 13a and 13b request that object 04 be provided to them, client 13a may actually receive object 04a and client 13b may receive object 04b, despite the fact that the reference 04 was a common global object identification."); *id.* at Abstract ("The defined order of the sequence of objects is determined from a client request for data."), 3:25-32, 9:32-10:4.

Otherwise, requesting objects with bandwidth greater than the network bandwidth would introduce latency (or choppiness) in contrast to Kliger's purpose. *Id.* at 5:26-28 ("The selection of objects may depend upon client parameters such as observed network latency in real time or desired object quality."); *id.* at claim 3 (". . . for each of the certain objects, in one of the server processor and requesting processor, determining one version of the object to be optimal for transmission in terms of available bandwidth of the communication channel; and using the determined version of the object in the set of objects transmitted in the data stream."); *id.* at 10:1-4 ("Quality may also have other meaning such as bandwidth . . .").

In fact, the PTAB previously found a similar system (Carmel)—in which different bandwidth files are selected based on network bandwidth—to Kliger's bandwidth scalability to disclose sending faster than the playback rate. WebPower Final Written Decision on Remand (Ex. 12), at 21-23 ("Carmel describes client 30 making an assessment of the rate of data transfer over the link from the server and, if necessary, changing the quality level accordingly. . . . Petitioner reasons that [b]y the client selecting a lower quality level for the slices, each slice can be individually transmitted faster . . . . We accordingly find that Petitioner sufficiently identifies disclosure in Carmel that meets the last limitation of claim 10 under Carmel's lag-recovery mechanism.") (alteration in original, citations and quotations omitted). Thus, Kliger also discloses sending data faster than the playback rate like Carmel.

Accordingly, Kliger teaches claim 4.

I. SNQP9: Imai and Carmel, Glaser, or TCP render obvious claims 1 and 4

1. Motivation to Combine

A POSITA would have found it obvious to combine the teachings of Carmel or Glaser and Imai because the combination is merely a matter of applying a known technique to a known device for improvement to yield predictable results, including for claim 4.

Imai and Carmel disclose similar client-pull systems, including requesting data by serial number using a file list or index file. *See supra* 12-13, 14; claim 1[a]-[c] below. A POSITA under *KSR* (*see supra* 20-21) at the time of the purported invention was made would have been motivated to combine the teachings of Imai and Carmel given the similarity of the Imai and Carmel and a POSITA would have been motivated to review other client-pull systems to further efficiency and enhance Imai's functionality.

A POSITA would have been further motivated to implement Carmel's sending faster than playback rate and would have had a reasonable expectation of success in doing so considering Imai's existing server and client functionality. Imai already describes a cache region used to protect against network bandwidth changes and could store the data transmitted faster than the playback rate. Imai (Ex. 7), at 21:5-18 ("This cache region 121a is utilized in order to relax the network congestion, and when a display request for the file stored in the cache region 121a is issued from the user, basically, the access to the file server 110 is not made and the corresponding file is read out from the cache region 121a and displayed on a screen") Thus, Imai could implement Carmel's sending faster than playback rate seamlessly.

Also, a POSITA would have been motivated to combine the references given the patentee's admissions that the TCP protocol was well-known for proper data delivery from server to client and that media players, including client buffers, were well-known for playback. *See supra* 2-3; *Unwired Planet*, 841 F.3d at 1003; *McCoy*, 850 F. App'x at 789. A POSITA would have been motivated to improve on this known knowledge, *e.g.*, transmitting data and storing such data in the client's buffer, by combining Carmel's transmitting faster than the playback rate to further enhance Imai's efficiency in buffering and playing streaming data. *Advanced Micro Devices*, 848 F.2d at 1570; *WesternGeco*, 889 F.3d at 1308. Accordingly, a POSITA would have found it obvious to combine the teachings of Imai and Carmel.

Additionally, a POSITA would have been motivated to combine Imai and Glaser. Imai and Glaser both teach video distribution systems. *See supra* 14-15; claim 1[a]-[c] below.

A POSITA would have been motivated to implement Glaser's sending faster than the playback rate in Imai to enhance Imai's functionality. Also, Glaser is cited on the front page of the '011 Patent, so a POSITA would have been motivated to review and apply it to other prior art,

including Imai. A POSITA would have been further motivated to implement such function considering Imai's existing server structure and buffering ability to store the data sent faster than the playback rate. Imai (Ex. 7), at Abstract, 10:15-25, 11:5-16, 21:32-43. Accordingly, a POSITA would have found it obvious to combine the teachings of Imai and Glaser.

2. **Claim 1**

(a) ***1[a] - A media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number, comprising***

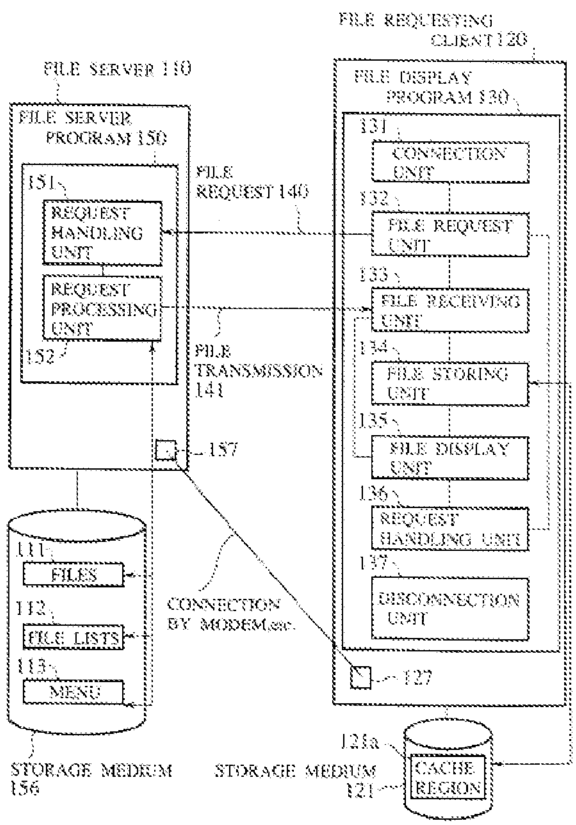
Imai teaches a media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number.

Imai teaches that a media player receives video or audio files over the Internet from a media source or server. Imai (Ex. 7), at Abstract ("A scheme for transferring files from a file server to a file requesting client, which enables request and transfer of files which are related to a user requested file at a time of transferring a user requested file."); *id.* at 1:28-31 ("file is used as a generic term for all the information which is formed by media such as text, figures, static images, video, audio, etc., and which can be electronically provided. One file may be formed by multiple media."); *id.* at 12:9-20 ("The viewer programs include those which are equipped with a processing system for interpreting and executing a program transmitted from the file server 110. For example, using the viewer equipped with a function for interpreting and executing the Java language, it is possible to realize the file transfer method of this first embodiment by receiving from the file server 110 a program which contains the network request processing function, the file receiving

processing function, the file storing processing function and the file display processing function, which are described in the Java language, and executing this program at the viewer side."); *id.* at 20:35-43 ("This information communication system is a client-server system for providing various files (hypertext in HTML format, static images, video images, audio data, etc.) from a server to a client by utilizing the network environment such as that of the Internet, for example. This system of FIG. 23 generally comprises a file server 110 which operates as a WWW server for providing files, and a file requesting client 120 which operates as a WWW client for requesting files."); *id.* at 20:58-60 ("On the file requesting client 120, the file display program 130 is operating. This file display program 130 functions as the WWW client program (WWW browser) . . . .")

The media player and server, each with various components, are shown in Figure 1 below:

FIG.1



Id. at Fig. 1; *id.* at 12:21-24 ("FIG. 8 shows . . . a list of files that can be provided by the file server 110 for a case of realizing the file transfer method of this first embodiment by utilizing the Java program (called applet)."); *id.* at 20:65-21:4 ("The connection unit 131 is a program for carrying out a connection to the file server 110 by the protocol such as TCP/IP. The file request unit 132 is a program for issuing the file transfer request to the file server 110 by the file transfer protocol such as HTTP. The file receiving unit 133 is a program for receiving a file provided from the file server 110.")

Imai teaches playing each file, where each file is associated with a serial number (URL or filename). *Id.* at 22:11-21 ("the user or program requests the transfer and the display of the file by issuing the file transfer request (step S11). For this purpose, it suffices to use the URL which is an identifier for uniquely identifying the file. For example, when the URL is specified as 'http://isl.rdc.tochina.co.jp/index.html', it is possible to uniquely identify the file for which the transfer protocol is the HTTP . . . . it is assumed that the requested file is a file-A which is a file identified by such an identifier A."); *id.* at 22:39-43 ("In the file list 2500 of FIG. 25, <LIST> is a keyword indicating that this file is the file list. The following character string indicates an identifier of the file-A which is requested by the user. The second and subsequent lines indicate the identifiers of the related files of this file-A."); *id.* at 22:53-59 ("Then, the file list is composed using the identifiers of the related files extracted from this file-A (step S16). After that, at the steps S17 to S21, the multiple files request transmission processing to take out the identifier of the file from the produced file list one by one and sequentially issue the file request for each related file is started.")

Accordingly, Imai teaches claim 1[a]. Further, Imai especially teaches 1[a] if WAG's proposed plain and ordinary meaning construction of "serial number / identifier" is adopted. As

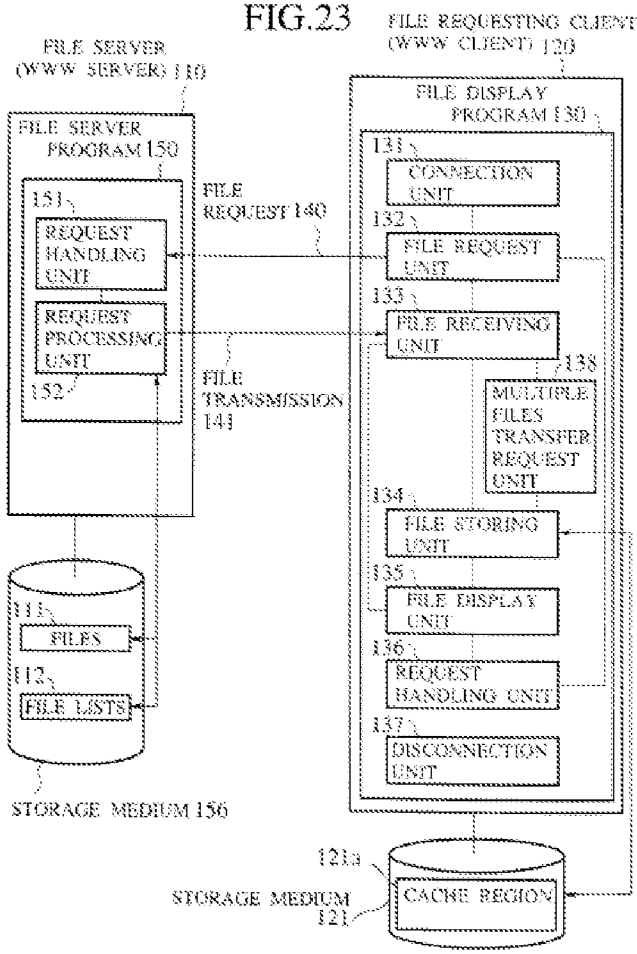
discussed previously, WAG contends that "serial number/identifier" is not limited to "consecutive [numbers] or limited to numerals," and thus can broadly encompass several forms. *See supra* 18. Imai teaches that the requested files are associated with a filename or other identifier and is not limited to any particular form. Imai (Ex. 7), at Fig. 4, 7:49-51, 22:11-21, 22:39-47, 22:53-62. Thus, Imai especially anticipates under WAG's broad plain and ordinary meaning construction.

(b) ***1[b] - a processor; a memory; a connection to the network; and***

Imai teaches processor, memory, and a connection to the network. Imai (Ex. 7), at 20:58-64 ("On the file requesting client 120, the file display program 130 is operating. This file display program 130 functions as the WWW client program (WWW browser), and includes a connection unit 131, a file request unit 132, a file receiving unit 133, a file storing unit 134, a file display unit 135, a request handling unit 136, a disconnection unit 137, and a multiple files transfer request unit 138."); *id.* at 21:5-7 ("The file storing unit 134 is a program for caching (temporarily storing) a file received from the file server 110 into a cache region 121a of a storage medium 121."); *id.* at 7:17-19 ("The file requesting client 120 is connected with a storage medium 121 having a cache region 121a in order to store the files 111 received from the file server 110."); *id.* at 20:35-43 ("This information communication system is a client-server system for providing various files (hypertext in HTML format, static images, video images, audio data, etc.) from a server to a client by utilizing the network environment such as that of the Internet, for example. This system of FIG. 23 generally comprises a file server 110 which operates as a WWW server for providing files, and a file requesting client 120 which operates as a WWW client for requesting files.")

Figure 23 below outlines a client device including the processor, memory, and connection to the network:

FIG.23



Id. at Fig. 23; *see also id.* at Fig. 1.

Accordingly, Imai teaches claim 1[b].

(c) ***1[c] - media player software comprising instructions to cause the media player to request from the media source a predetermined number of media data elements;***

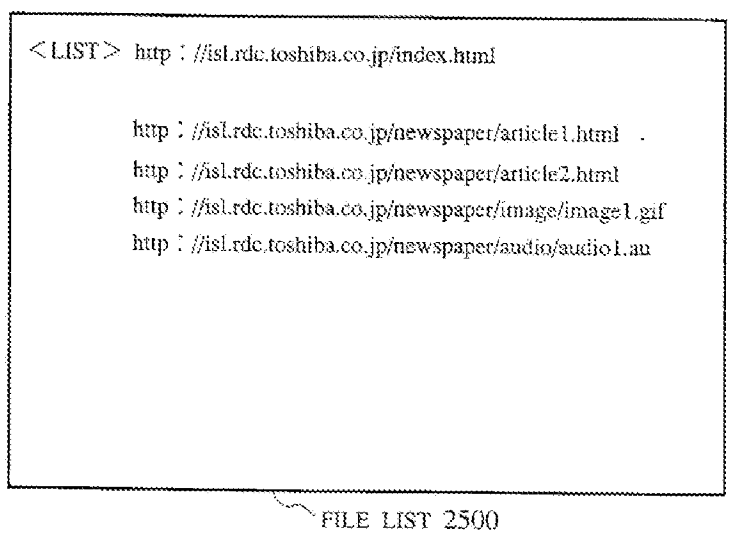
Imai teaches the media player to request from the media source a predetermined number of media data elements.

As discussed in claim 1[a]-[b], Imai teaches a media playing device. *See also* Imai (Ex. 7), at 12:9-20 ("The viewer programs include those which are equipped with a processing system for interpreting and executing a program transmitted from the file server 110. For example, using

the viewer equipped with a function for interpreting and executing the Java language, it is possible to realize the file transfer method of this first embodiment by receiving from the file server 110 a program which contains the network request processing function, the file receiving processing function, the file storing processing function and the file display processing function, which are described in the Java language, and executing this program at the viewer side.")

Imai further teaches the media player requests a predetermined number of media data elements because it will request and buffer multiple files by serial number from a file listing. Imai describes that a server provides a file list with a listing of available files by identifier or serial number for download. A file listing is shown below in Figure 25:

FIG.25



Id. at Fig 25; *see also id.* at Figs. 4, 6; *id.* at 22:39-47 ("In the file list 2500 of FIG. 25, <LIST> is a keyword indicating that this file is the file list. The following character string indicates an identifier of the file-A which is requested by the user. The second and subsequent lines indicate

the identifiers of the related files of this file-A. . . . [A] file having a name which ends with '.au' is an audio file."); *id.* at 10:15-19 ("The file list may be produced by a file provider, or produced automatically by extracting link information from the requested file. In a case of automatically producing the file list, a program for automatically producing the file list is stored in the file server 110 in advance."); *id.* at 1:28-31 ("file is used as a generic term for all the information which is formed by media such as text, figures, static images, video, audio, etc., and which can be electronically provided. One file may be formed by multiple media.").

Imai requests and buffers by serial number (URL or filename) multiple or predetermined files from the file list. *Id.* at 22:55-59 ("at the steps S17 to S21, the multiple files request transmission processing to take out the identifier of the file from the produced file list one by one and sequentially issue the file request for each related file is started."); *Id.* at 23:28-45 ("When the transfer interrupt condition does not hold, the multiple files transfer request unit 138 selects one not yet transferred file from the files listed in the file list, and gives its identifier to the file request unit 132 to command the issuance of the file transfer request (step S18). When the transfer target file is selected in this manner (step S19 YES), the file request unit 132 transmits the file transfer request for the specified file in the HTTP format to the file server program 150 (step S20). The file server program 150 then transfers the file specified by the file transfer request from the file request unit 132 to the file requesting client 120. This file is received by the file receiving unit 133, and stored into the cache region 121a by the file storing unit 134 (step S21). In a case where the file received at the step S13 is the file list, the file received first at the step S21 is going to be the file-A, and in such a case, the screen display of the file-A is also carried out by the file display unit 135."); *id.* at 23:46-49 ("The processing of the steps S17 to S21 is repeatedly executed until there is no more not yet transferred file among the files listed in the file list, and every time one file

transfer is carried out . . . ."); *id.* at 21:32-35 ("The multiple files transfer request unit 138 is a program for commanding the file request unit 132 to issue multiple file transfer requests with respect to one transfer request handled by the request handling unit 136. . . ."); *id.* at 2:33-38 ("In order to provide necessary files to a user immediately in response to a request from a user, it is necessary to transfer multiple files collectively from the server to the client while the client computer is connected with the network, and store the transferred files at the client side."). The foregoing is also shown in Figure 24.

Accordingly, Imai teaches claim 1[c]. Further, Imai especially teaches claim 1[c] if WAG's construction of "predetermined" discussed with reference to claim 1[g] as any specified data is applied here. *See supra* 18-19. Imai describes requesting multiple or predetermined files from the file list by serial number. *See* above. Thus, Imai especially anticipates under WAG's broad plain and ordinary meaning construction given specified data is requested.

(d) ***1[d] - instructions to cause the media player to receive media data elements sent to the media player by the media source and store the media data elements in the memory;***

Imai teaches that the media player receives and stores media data elements in memory. As described in claim 1[b]-[c], Imai teaches a media playing device with a cache region to store the files received. *See also* Imai (Ex. 7), at 21:5-13 ("The file storing unit 134 is a program for caching (temporarily storing) a file received from the file server 110 into a cache region 121a of a storage medium 121. . . . A stored position and a file name of a file cached into the cache region 121a are independently managed by the file display program 130."); *id.* at 11:5-17 ("according to the content of the received file list, a file is requested from the file request unit 402, and the file requested by the file request unit 402 is received at the file receiving unit 403 and the received file is stored into the cache region 121a of the storage medium 121 by the file storing unit 404. In

addition, the file corresponding to the user's request is displayed on a viewer by the file display unit 405. The operations of the file request unit 402, the file receiving unit 403, and the file storing unit 404 according to the file list are repeated as many times as a number of files listed in the file list."); *id.* at 14:1-7 ("By storing the requested file and the related files transferred from the file server in response to the request into the storage medium at the file requesting client side, it becomes possible to refer to not just the requested file but also its related files even when the connection with the file server is disconnected.").

Accordingly, Imai teaches claim 1[d].

(e) ***1[e] - instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer;***

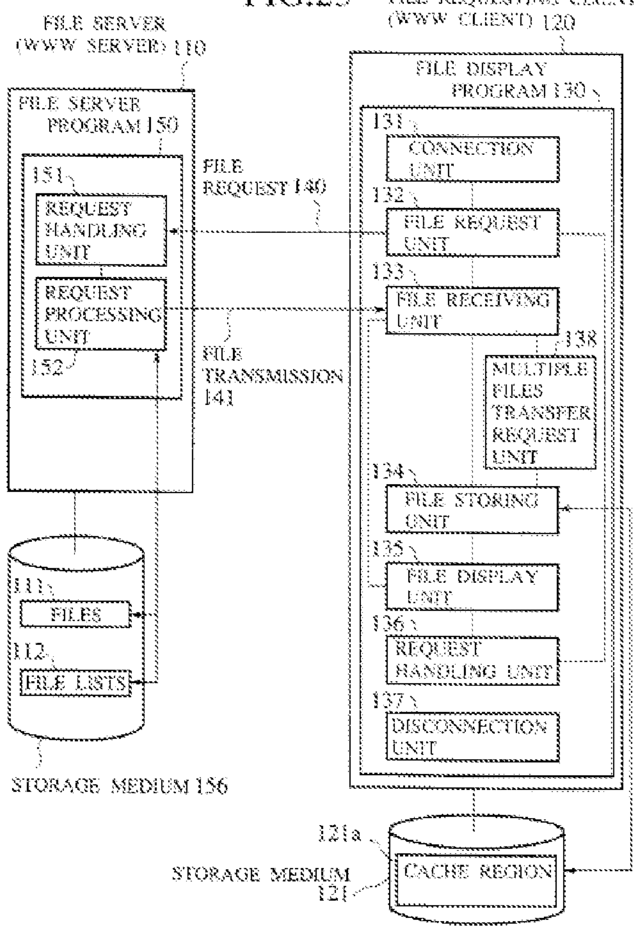
Imai teaches a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer.

Imai teaches a player buffer, *e.g.*, file storing unit and/or file display program (*e.g.*, browser), which manage the storing of received files to the cache region. Imai (Ex. 7), at 20:58-64 ("On the file requesting client 120, the file display program 130 is operating. This file display program 130 functions as the WWW client program (WWW browser), and includes a connection unit 131, a file request unit 132, a file receiving unit 133, a file storing unit 134, a file display unit 135, a request handling unit 136, a disconnection unit 137, and a multiple files transfer request unit 138."); *id.* at 21:5-13 ("The file storing unit 134 is a program for caching (temporarily storing) a file received from the file server 110 into a cache region 121a of a storage medium 121. . . . A stored position and a file name of a file cached into the cache region 121a are independently

managed by the file display program 130."); *id.* at 7:47-49 ("This received file is then stored into the cache region 121a of the storage medium 121 by the file storing unit 134.").

The relationship between the 134 "File Storing Unit" and 121a "Cache Region" is shown in Figure 23 below:

FIG.23



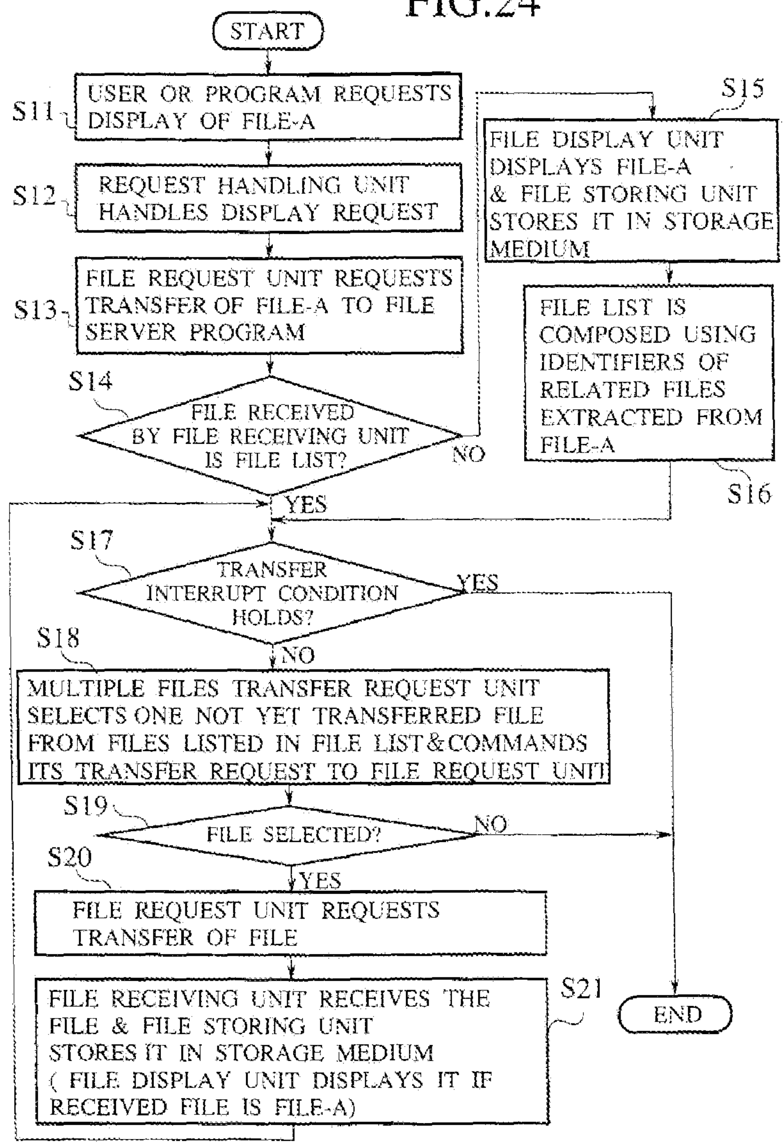
Id. at Fig. 23.

Imai teaches the file storing unit and/or file display program (*e.g.*, browser) maintain a record of the serial number (URL or filename) of the last file received because Imai requests and stores each file associated with a serial number from the file list.

As described in claim 1[c], the media player will issue requests for each file by serial number from the file list. Each file with a serial number will be stored in the cache region, including the last one received. *Id.* at 21:5-11 ("The file storing unit 134 is a program for caching (temporarily storing) a file received from the file server 110 into a cache region 121a of a storage medium 121. The file cached into the cache region 121a is automatically allocated with a name and stored without a command from the user, unlike a usual file which is stored using a file name explicitly specified by the user."); *id.* at 23:29-41 ("the multiple files transfer request unit 138 selects one not yet transferred file from the files listed in the file list, and gives its identifier to the file request unit 132 to command the issuance of the file transfer request (step S18). . . . The file server program 150 then transfers the file specified by the file transfer request from the file request unit 132 to the file requesting client 120. This file is received by the file receiving unit 133, and stored into the cache region 121a by the file storing unit 134 (step S21)."); *id.* at 2:33-38 ("In order to provide necessary files to a user immediately in response to a request from a user, it is necessary to transfer multiple files collectively from the server to the client while the client computer is connected with the network, and store the transferred files at the client side.")

The foregoing requesting and storing of the files from the file list, including the last one received in shown in Figure 24 below, particularly S21 "File Receiving Unit Receives the File & File Storing Unit Stores it in Storage Medium . . .":

FIG.24



Accordingly, Imai teaches claim 1[e].

(f) ***1[f] - instructions to cause the media player to play media data elements sequentially from the player buffer; and***

Imai teaches in claim 1[a]-[b] a media player device to play each requested file. *See also* Imai (Ex. 7), at 12:9-20 ("The viewer programs include those which are equipped with a processing system for interpreting and executing a program transmitted from the file server 110. For example, using the viewer equipped with a function for interpreting and executing the Java language, it is

possible to realize the file transfer method of this first embodiment by receiving from the file server 110 a program which contains the network request processing function, the file receiving processing function, the file storing processing function and the file display processing function, which are described in the Java language, and executing this program at the viewer side."); *id.* at 20:58-60 ("On the file requesting client 120, the file display program 130 is operating. This file display program 130 functions as the WWW client program (WWW browser) . . . .")

Imai further teaches playing each file sequentially from the player buffer. *Id.* at 11:5-17 ("Then, according to the content of the received file list, a file is requested from the file request unit 402, and the file requested by the file request unit 402 is received at the file receiving unit 403 and the received file is stored into the cache region 121a of the storage medium 121 by the file storing unit 404. In addition, the file corresponding to the user's request is displayed on a viewer by the file display unit 405. The operations of the file request unit 402, the file receiving unit 403, and the file storing unit 404 according to the file list are repeated as many times as a number of files listed in the file list."); *id.* at 21:25-29 ("The file display unit 135 is a program for displaying the requested file on a screen. The request handling unit 136 is a program for handling requests for file transfer or file display from the user or program."); *id.* at 22:11-21 ("the user or program requests the transfer and the display of the file by issuing the file transfer request (step S11). For this purpose, it suffices to use the URL which is an identifier for uniquely identifying the file. . . . [I]t is assumed that the requested file is a file-A which is a file identified by such an identifier A.").

Accordingly, Imai teaches claim 1[f].

(g) ***1[g] - instructions to cause the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.***

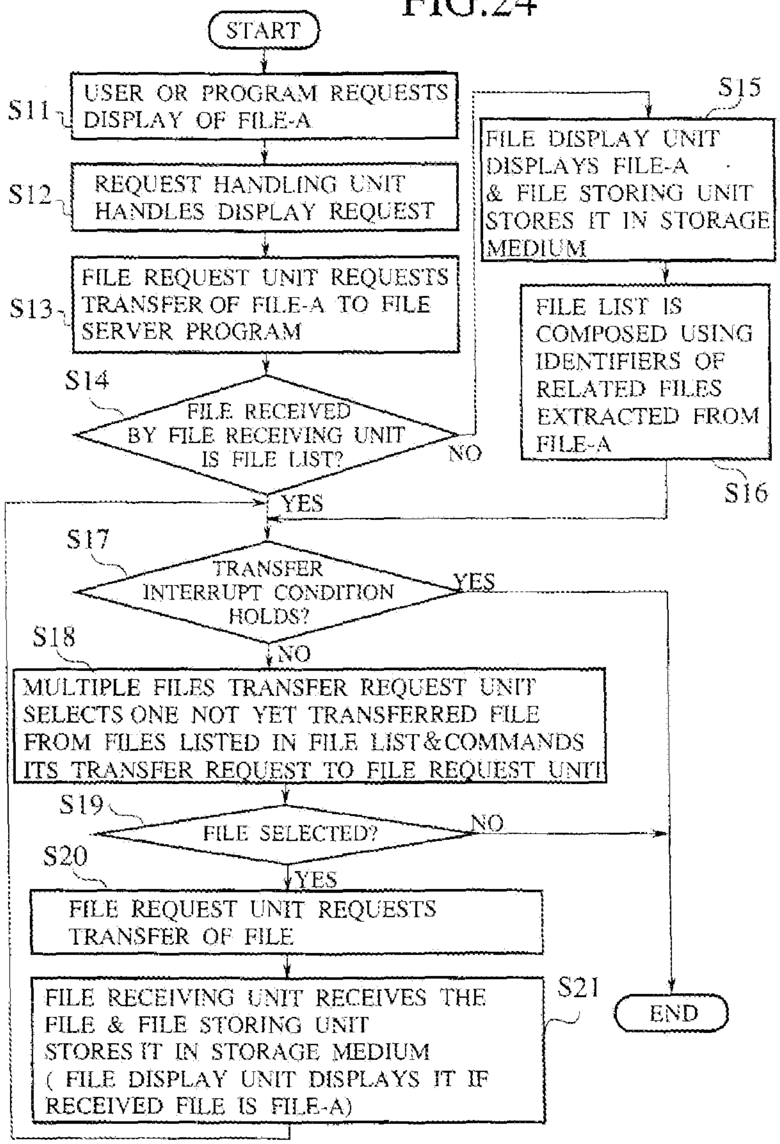
Imai teaches that the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.

As discussed in claim [c], Imai requests multiple or predetermined files from the file list by serial number (URL or filename). Specifically, Imai requests each file by serial number (URL or filename) from the file list and repeats those requests to maintain or store all such multiple files for the duration of the program. *Id.* at 23:29-50 ("the **multiple files transfer request unit 138 selects one not yet transferred file from the files listed in the file list**, and **gives its identifier to the file request unit 132 to command the issuance of the file transfer request (step S18)**. . . . The file server program 150 then transfers the file specified by the file transfer request from the file request unit 132 to the file requesting client 120. **This file is received by the file receiving unit 133, and stored into the cache region 121a by the file storing unit 134 (step S21)** . . . . The **processing of the steps S17 to S21 is repeatedly executed until there is no more not yet transferred file among the files listed in the file list**, and every time one file transfer is carried out, whether the transfer interrupt condition holds or not is judged.") (emphasis added).

The requesting and storing (or buffering) by serial number (URL or filename) of multiple or predetermined files from the file list is shown in Figure 24 below, particularly S18 "Multiple

Files Transfer Request Unit Selects One Not Yet Transferred File from Files Listed in File List & Commands its Transfer Request to File Request Unit" and S21 "File Receiving Unit Receives the File & File Storing Unit Stores it in Storage Medium . . .":

FIG.24



Thus, Imai teaches claim limitation 1[g]. Further, Imai especially teaches claim 1[g] if WAG's proposed plain and ordinary meaning construction of the predetermined limitation is adopted. As discussed previously, WAG broadly contends that any specified data in the client

buffer at any point (whether before the client connects to the server or after the client connects to the server) would cover this claim limitation. *See supra* 18-19 (WAG contending predetermined is a "specified number of media data elements" that "could be determined after connection to the network and after connection to the media source" and are not constant or the "same number").

Imai requests and buffers by serial number (URL or filename) multiple or predetermined files from the file list. Imai (Ex. 7), at 22:53-62, 23:28-54, Figs. 23-24. Thus, Imai especially anticipates under WAG's broad plain and ordinary meaning construction given the client buffers specified data.

3. **Claim 4**

Claim 4 states, "4. The media player of claim 1, wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player."

As discussed above, a POSITA would have been motivated to combine Imai and Carmel for claim 4. Imai describes the client receiving a predetermined number of files. *See* claim 1[g]. Carmel teaches that the media player receives the media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player. *See supra* 43-46. Accordingly, Imai and Carmel render obvious claims 1 and 4.

Additionally, Imai and Glaser or TCP teach claim 4. A POSITA would have been motivated to combine Imai and Glaser or TCP for claim 4. Imai describes the client receiving a multiple or predetermined files from the file list by serial number (URL or filename. *See* claim

1[g]. Glaser or TCP teaches receiving data at a rate faster than the playback rate. *See supra* 47. Accordingly, Imai and Glaser or TCP render obvious claims 1 and 4.

IX. TIMING FOR REQUESTING *EX PARTE* REEXAMINATION

The filing of this Request for *Ex Parte* Reexamination is timely. The '011 Patent expired (March 28, 2021) less than 6 years from the date of filing this Request for *Ex Parte* Reexamination. 35 U.S.C. §§ 286, 302; See M.P.E.P. § 2211.

X. COMMUNICATIONS TO THIRD PARTY REQUESTERS

The Requesters' correspondence address is:

Frank M. Gasparo
Attorney for Requesters
VENABLE LLP
1270 Avenue of the Americas, 24th Floor
New York, NY 10020
T 212-370-6273
F 212-307-5598
fmgasparo@Venable.com

XI. CERTIFICATION THAT REEXAMINATION IS NOT PROHIBITED

In accordance with 37 C.F.R. § 1.510(b)(6), Requesters hereby certify that the statutory estoppel provisions of 35 U.S.C. § 315(e)(1) and 35 U.S.C. § 325(e)(1) do not prohibit the Requesters from filing this *Ex Parte* Reexamination Request.

Requesters, Duodecad IT Services Luxembourg S.à r.l., and others previously filed an IPR on the '011 Patent, but it was denied institution. Decision Denying Institution of *Inter Partes* Review, *FriendFinder Networks Inc. et al. v. WAG Acquisition L.L.C.*, IPR2015-01033, Paper No. 8 (Oct. 19, 2015). Thus, there is no estoppel because there was no final written decision on the claims challenged here.

> Section 315 explicitly limits the estoppel to the claims previously challenged and for those proceedings that resulted in a final written

> decision: 'The petitioner in an inter partes review *of a claim* in a patent under this chapter that *results in a final written decision* under section 318(a) . . . may not request or maintain a proceeding before the Office *with respect to that claim* on any ground that the petitioner raised or reasonably could have raised during that inter partes review.' § 315(e)(1) (emphases added). Because claim 7 was not at issue in the Apple IPR, the plain language of the statute supports the conclusion that Facebook is not estopped from challenging this claim in this proceeding, regardless of its dependency on claim 1. Moreover, as a joined party in the Apple IPR, Facebook could not have raised any grounds for the patentability of claim 7 because claim 7 was not already challenged in the proceeding.

Uniloc 2017 LLC v. Facebook Inc., 989 F.3d 1018, 1030-31 (Fed. Cir. 2021) (emphasis in original, citations omitted); 35 U.S.C. § 315(e).

Additionally, WebPower, Inc. previously filed an IPR on the '011 Patent, but it was denied institution. Decision Denying Institution of *Inter Partes* Review, *WebPower v. WAG Acquisition L.L.C.*, IPR2016-01161, Paper No. 7 (Dec. 12, 2016). Thus, there is no estoppel because there was no final written decision on the claims challenged here. *Uniloc*, 989 F.3d at 1030-31; 35 U.S.C. § 315(e).

Lastly, I.M.L. SLU previously filed an IPR on the '011 Patent, but it was denied institution. Decision Denying Institution of *Inter Partes* Review, *I.M.L. SLU v. WAG Acquisition L.L.C.*, IPR2016-01655, Paper No. 11 (Feb. 27, 2017). Thus, there is no estoppel because there was no final written decision on the claims challenged here. *Uniloc*, 989 F.3d at 1030-31; 35 U.S.C. § 315(e).

Accordingly, Requesters are not statutorily estopped from filing this request for *ex parte* reexamination.

XII. CONCLUSION

For at least the reasons above, Requesters respectfully request that claims 1 and 4 of the ’011 Patent be reexamined and declared unpatentable. If necessary, please charge any required fees that are not included herewith to our Deposit Account No. 22-0261.

Requesters respectfully request that this Request for *Ex Parte* Reexamination of the ’011 Patent be granted by the USPTO, and that the reexamination be conducted with “special dispatch” and with “priority over all other cases” in accordance with M.P.E.P. § 2261.

Dated: August 25, 2021

Respectfully submitted,

By: /Frank Gasparo/

Frank M. Gasparo (Reg. No. 44,700)
VENABLE LLP
1270 Avenue of the Americas, 24th Floor
New York, NY 10020
T 212-370-6273
F 212-307-5598
fmgasparo@Venable.com

Attorney for Requesters

Exhibit 1
to
Ex Parte Reexamination
of
U.S. Patent No.
8,327,011

US008122141B2

(12) **United States Patent**
Price

(10) **Patent No.: US 8,122,141 B2**
(45) **Date of Patent: *Feb. 21, 2012**

(54) **STREAMING MEDIA BUFFERING SYSTEM**

(75) Inventor: **Harold Edward Price**, Bethel Park, PA (US)

(73) Assignee: **WAG Acquisition, LLC**, Flanders, NJ (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **12/800,152**

(22) Filed: **May 10, 2010**

(65) **Prior Publication Data**

US 2010/0235536 A1 Sep. 16, 2010

Related U.S. Application Data

(63) Continuation of application No. 10/893,814, filed on Jul. 19, 2004, now Pat. No. 7,716,358, which is a continuation-in-part of application No. 09/819,337, filed on Mar. 28, 2001, now Pat. No. 6,766,376.

(60) Provisional application No. 60/231,997, filed on Sep. 12, 2000.

(51) **Int. Cl.**
G06F 15/16 (2006.01)

(52) **U.S. Cl.** **709/231**

(58) **Field of Classification Search** 709/230, 709/231
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,526,353	A	6/1996	Henley et al.	370/60.1
5,793,980	A	8/1998	Glaser et al.	395/200.61
5,822,537	A	10/1998	Katseff et al.	395/200.61
5,922,048	A	7/1999	Emura	709/219
5,923,655	A	7/1999	Veschi et al.	370/394
5,928,330	A *	7/1999	Goetz et al.	709/231
5,999,525	A	12/1999	Krishnaswamy et al.	370/352
6,002,720	A	12/1999	Yurt et al.	375/240
6,014,693	A	1/2000	Ito et al.	709/219
6,014,694	A	1/2000	Aharoni et al.	709/219
6,014,706	A	1/2000	Cannon et al.	709/231
6,029,194	A	2/2000	Tilt	709/219
6,047,317	A	4/2000	Bisdikian et al.	709/217
6,057,832	A	5/2000	Lev et al.	345/327
6,061,731	A	5/2000	Blakeslee	709/231
6,061,732	A	5/2000	Korst et al.	709/231
6,065,050	A	5/2000	DeMoney	709/219
6,085,221	A	7/2000	Graf	709/202
6,292,834	B1	9/2001	Ravi et al.	

(Continued)

OTHER PUBLICATIONS

Salehi et al., "Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing," IEEE/ACM Transactions on Networking, vol. 6, Issue 4, p. 397-410, Aug. 1998.

(Continued)

Primary Examiner — Joseph Avellino

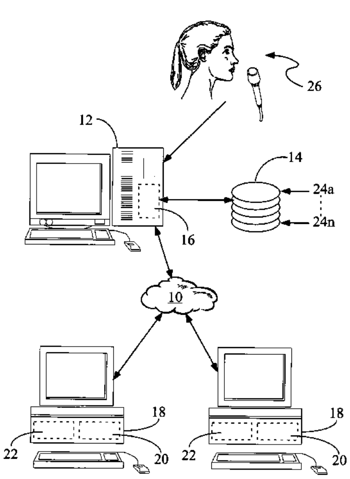
Assistant Examiner — Marshall McLeod

(74) *Attorney, Agent, or Firm* — Ernest D. Buff; Ernest D. Buff & Assoc. LLC; Gordon E. Fish

(57) **ABSTRACT**

Streaming media, such as audio or video files, is sent via the Internet. The media are immediately played on a user's computer. Audio/video data is transmitted from the server more rapidly than it is played out by the user system. The audio/video data in the user buffer accumulates; and interruptions in playback as well as temporary modem delays are avoided.

28 Claims, 3 Drawing Sheets



U.S. PATENT DOCUMENTS

6,588,015	B1	7/2003	Eyer et al.	
6,594,699	B1	7/2003	Sahai et al.	
6,598,228	B2	7/2003	Hejna	
6,637,031	B1 *	10/2003	Chou	725/87
6,665,751	B1	12/2003	Chen et al.	
6,708,213	B1	3/2004	Bommaiah et al.	
6,728,753	B1 *	4/2004	Parasnis et al.	709/203
6,829,368	B2	12/2004	Meyer et al.	
6,889,257	B1	5/2005	Patel	
6,907,481	B2	6/2005	Kovacevic	
7,054,500	B1	5/2006	Lillevold	
7,111,058	B1	9/2006	Nguyen et al.	
7,170,856	B1	1/2007	Ho et al.	
7,346,698	B2	3/2008	Hannaway	
7,373,413	B1	5/2008	Nguyen et al.	
2001/0047377	A1 *	11/2001	Sincaglia et al.	709/1

OTHER PUBLICATIONS

Rexford et al., "Smoothing variable-bit-rate video in an internetwork," IEEE/ACM Transactions on Networking, vol. 7, Issue 2, p. 202-215, Apr. 1999.

Bianchi, "Buffer sizing for high speed video information retrieval on ATM networks," Globecom—New York—, 1997, vol. 2, pp. 1057-1061.

* cited by examiner

Fig. 1

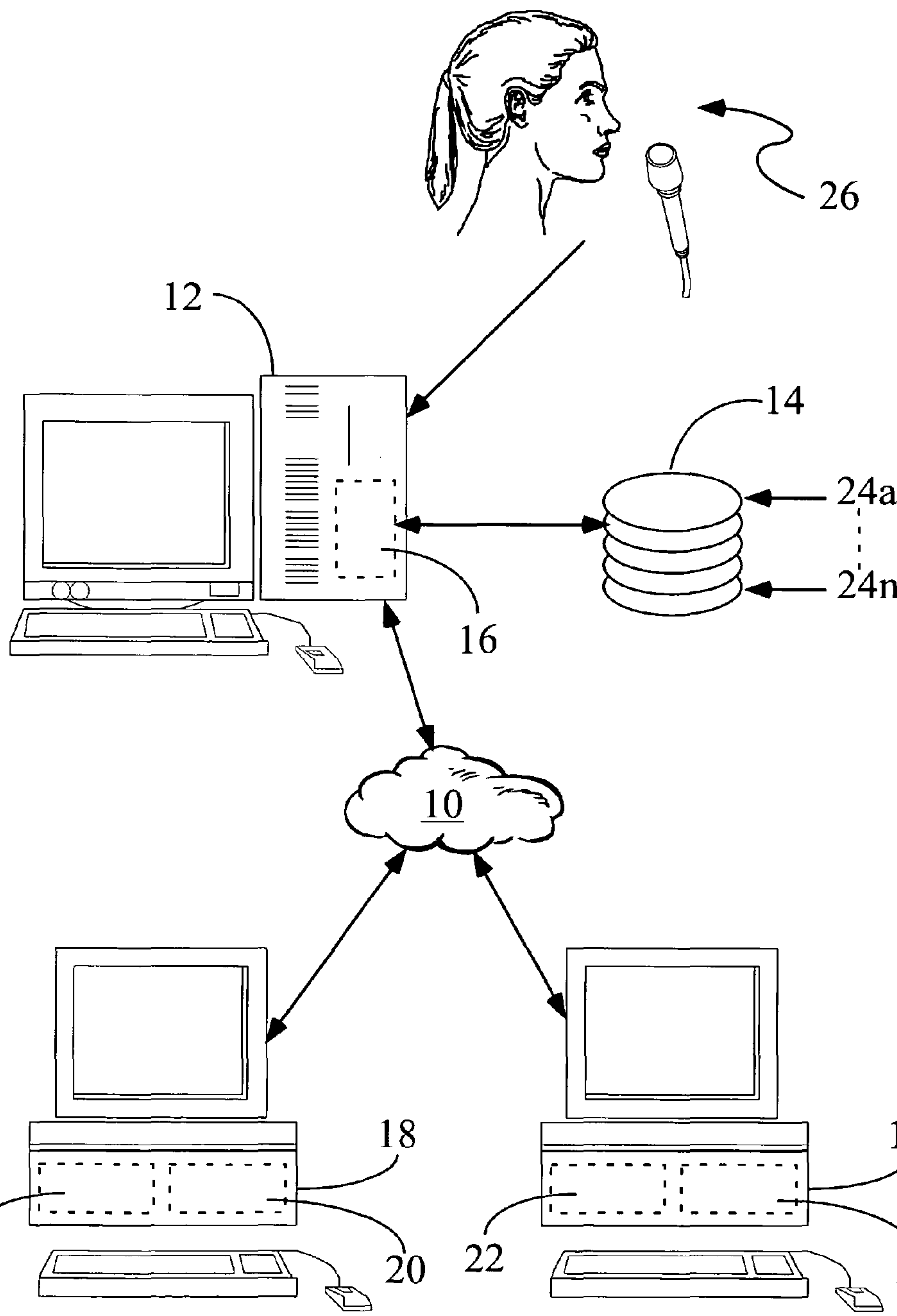


Fig. 2

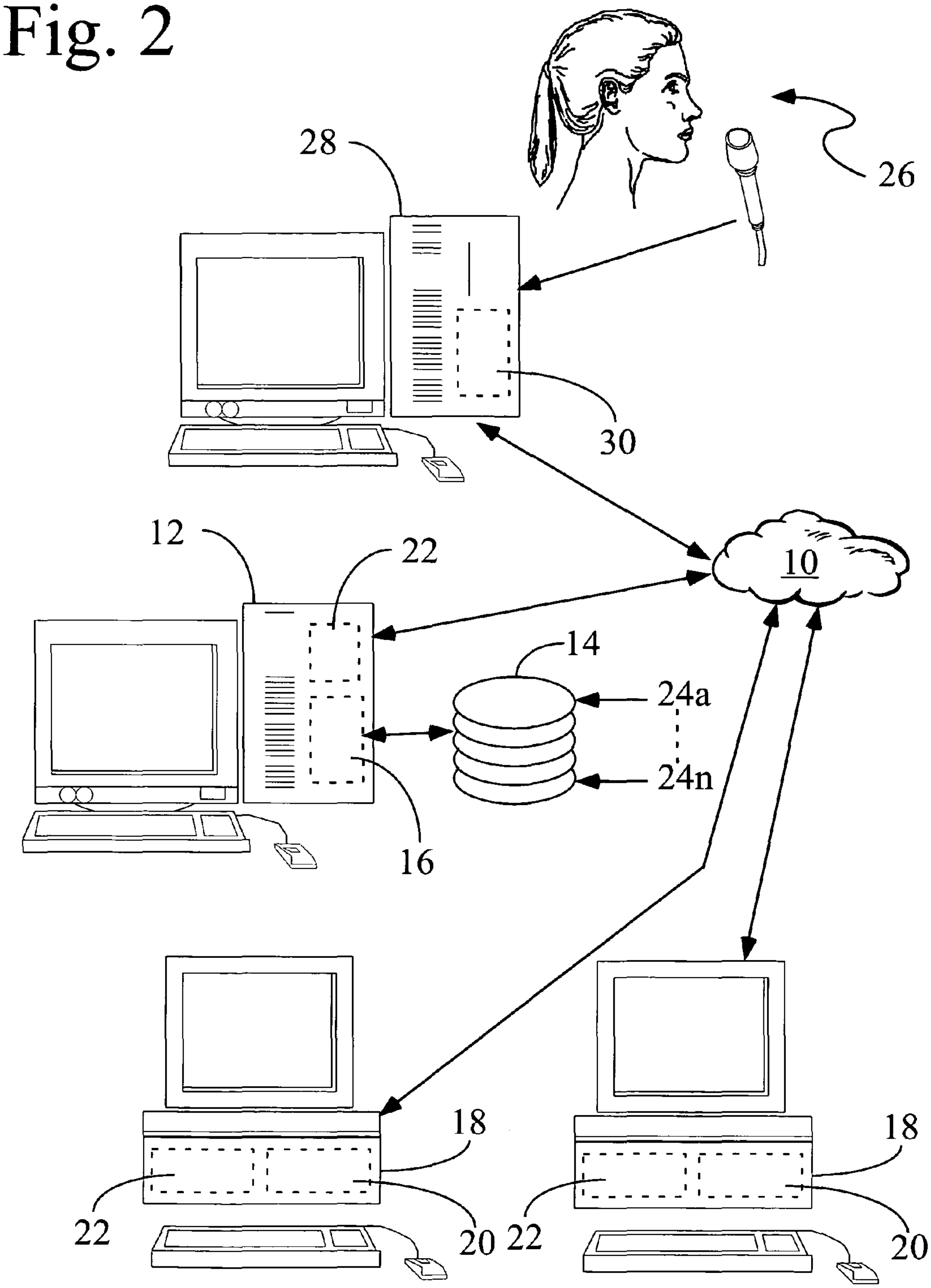
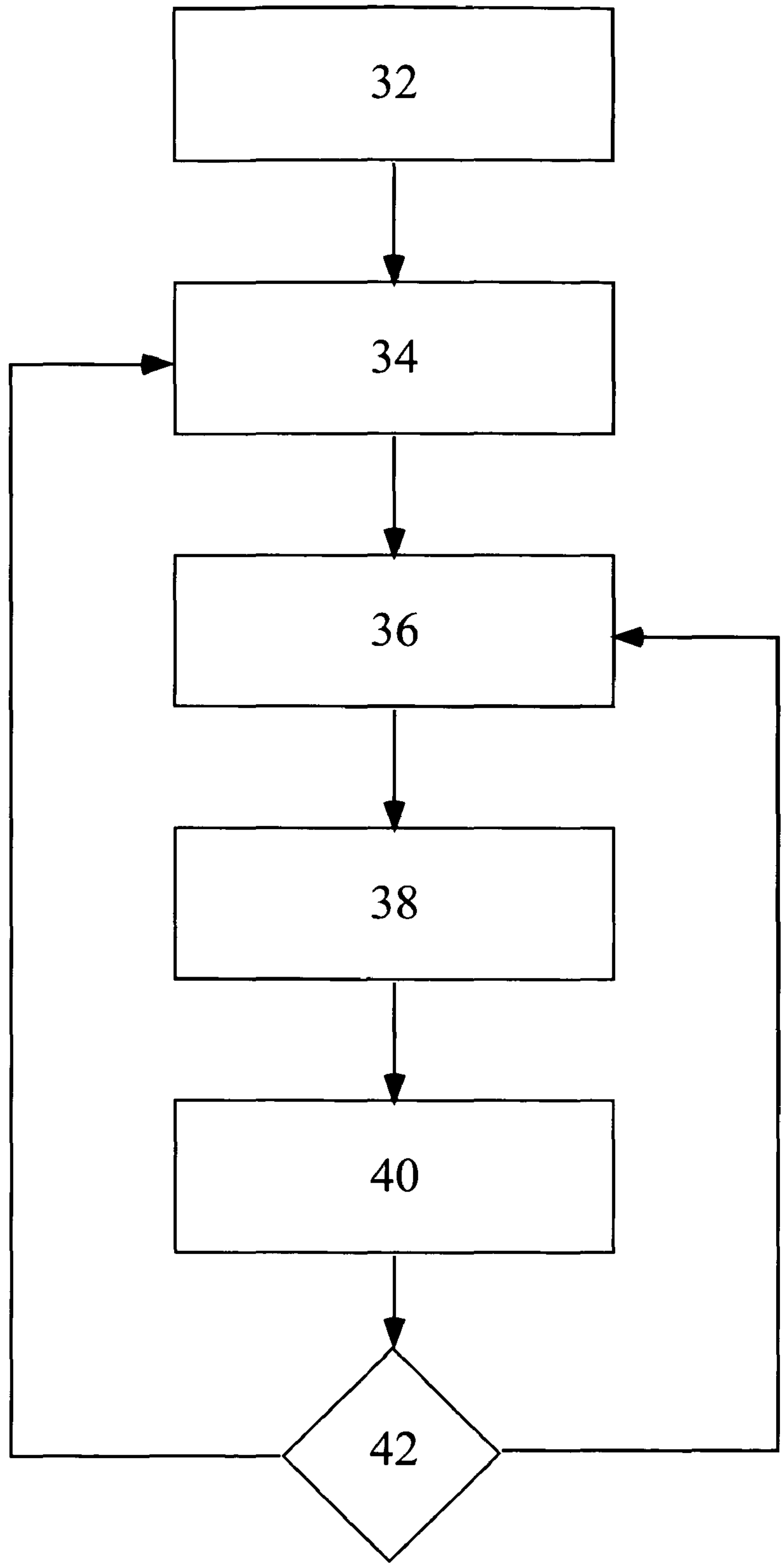


Fig. 3



STREAMING MEDIA BUFFERING SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 10/893,814, filed Jul. 19, 2004 (published on Dec. 9, 2004 as U.S. patent publication number 2004/0249969 A1, and now U.S. Pat. No. 7,716,358), which was a continuation-in-part of U.S. patent application Ser. No. 09/819,337, filed Mar. 28, 2001 (now U.S. Pat. No. 6,766,376), which claimed the benefit under 35 U.S.C. §119(e) of U.S. provisional patent application Ser. No. 60/231,997, filed Sep. 12, 2000; claims the benefit, under 35 U.S.C. §120, of the respective filing dates of said applications, as well as benefit of the filing date of copending U.S. patent application Ser. No. 10/825,869, filed Apr. 16, 2004 (published on Dec. 23, 2004 as U.S. patent publication number 2004/260828 A1), which was a continuation of said U.S. patent application Ser. No. 09/819,337, filed Mar. 28, 2001 (now U.S. Pat. No. 6,766,376), which claimed the benefit under 35 U.S.C. §119 (e) of said U.S. provisional patent application Ser. No. 60/231,997, filed Sep. 12, 2000; and hereby incorporates by reference the entire disclosure of each of said prior applications.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to multimedia computer communication systems; and more particularly, to a buffering system for streaming media, such as audio/video, on the Internet.

2. Description of the Related Art

Prior to the development of Internet streaming media technologies, audio and video were formatted into files, which users needed to download to their computer before the files could be heard or viewed. Real time, continuous media, as from a radio station, was not suitable for this arrangement in that a file of finite size must be created so it could be downloaded. The advent of streaming media technologies allowed users to listen or view the files as they were being downloaded, and allowed users to "tune-in" to a continuous media broadcast, or "stream", such as from a radio station. There are two fundamental types of streaming media: (i) material that originates from a source having a real-time nature, such as a radio or TV broadcast, and (ii) material that originates from a non-real-time source such as from a disk file. An example of non-real-time material might be a piece of music stored as a disk file, or a portion of a broadcast that originally was real-time, perhaps yesterday's TV evening news, and was recorded into a disk file. For purposes of clarity within this document, streaming media of type (i) will be referred to as "broadcast" media, and streaming media of type (ii) will be referred to as "file based" media. Broadcast streaming media has as its source a system or arrangement that by definition can only be transmitted to users as fast as the material is generated; for example, a disk jockey speaking into a microphone. Broadcast streaming media is the focus of this patent application.

Since audio and video media must play out over a period of time it is more appropriate to think of bandwidth requirements than file size. The bandwidth requirement of an audio or video media refers to the data rate in bits per second that must be transmitted and received in order to listen or view the material uninterrupted. Transmitting the audio or video material over a connection slower than the bandwidth requirement

results in unsatisfactory viewing or listening, if viewing or listening is possible at all. The connection available to most Internet users is by dial-up modem, which has a maximum receive data rate of 56,000 bits per second. Most audio and video available on the Internet has been compressed to be listenable or viewable within the 56,000 bits per second modem bandwidth. Requirements for achieving adequate audio and video over the Internet generally consume a considerable portion of the listener's available bandwidth.

Internet connection quality can vary rapidly over time, with two primary factors responsible for degradation of the instantaneous bandwidth actually available to the user. These factors are the quality of the user's modem connection over telephone lines, which can have periods of interference causing reduced available bandwidth, and momentary Internet congestion at various points along the route over which the user's data flows. Each of these factors can cause delays and interruptions in the transmission of data to the user. Internet data communications devices such as routers are designed to drop data "packets" if they get overloaded. For material that is not time sensitive, these dropped packets will usually be resent, and the user will eventually be presented with the material. However, since streaming media is time sensitive, dropped packets can have a significant impact on the receipt and playback of an audio or video stream. These degradations in the receipt of Internet data are very common, and prevent most users from being able to listen to or view streaming media without interruption unless some special provisions have been incorporated into the user's computer software to accommodate data transmission interruptions.

These interruptions are commonly referred to as "dropouts", meaning that the data flow to the user has been interrupted (i.e., the audio "drops out"). Dropouts can be extremely annoying—for example, while listening to music. The current state-of-the-art solution to the problem uses a pre-buffering technique to store up enough audio or video data in the user's computer so that it can play the audio or video with a minimum of dropouts. This process requires the user to wait until enough of the media file is buffered in memory before listening or viewing can begin. The media data is delivered by a server computer which has available to it the source of the media data, such as by a connection to a radio station. When the user connects to the server via the Internet, audio/video output at the user's system is delayed while the user's buffer is filled to a predetermined level. Typical pre-buffering wait times range from 10 to 20 seconds or more, determined by the vendor providing the audio or video media. Even with this pre-buffering process, interruptions in playback still occur.

In this process, the user has a software application on the computer commonly called a "media player". Using the features built into the media player, the user starts the audio or video stream, typically by clicking on a "start" button, and waits 10-20 seconds or so before the material starts playing. During this time data is being received from the source and filling the media player's buffer. The audio or video data is delivered from the source at the rate it is to be played out. If, for example, the user is listening to an audio stream encoded to be played-out at 24,000 bits per second, the source sends the audio data at the rate of 24,000 bits per second. Provided that the user waits 10 seconds, and the receipt of the buffering data has not been interrupted, there is enough media data stored in the buffer to play for 10 seconds.

Gaps in the receipt of audio/video data, due to Internet slowdowns, cause the buffer to deplete. Because transmission of audio/video media data to the user takes place at the rate it is played out, the user's buffer level can never be increased or

replenished while it is playing. Thus, gaps in the receipt of audio/video media data inexorably cause the buffer level to decrease from its initial level. In time, extended or repeated occurrences of these gaps empty the user's buffer. The audio/video material stops playing, and the buffer must be refilled to its original predetermined level before playing of the media resumes.

By way of illustration in a 10 second pre-buffering scenario, if the data reception stopped the instant that the media started playing, it would play for exactly 10 seconds. Once it starts playing, the media data plays out of the buffer as new media data replenishes the buffer. The incoming data rate equals the rate at which the data is played out of the user's buffer, assuming the receipt of data across the Internet is unimpeded. If there are no interruptions in the receipt of the media data for the duration of the time the user listens to or watches the material, the buffer level remains constant and there will still be 10 seconds of data stored in the media player's buffer when the user stops the player. On the other hand, if the media player encounters interruptions totaling 6 seconds while playing the material, there would only be 4 seconds of media data remaining in the buffer when the user stopped it. If data reception interruptions at any time during the playing exceed 10 seconds, the user's media player buffer becomes exhausted. There is no media data to play, and the audio or video stops—a dropout has occurred. At this point a software mechanism in the media player stops attempting to play any more of the material, and starts the buffering process again. The media player remains silent until the buffer refills, at which time the media player will once again start playing the material.

There are two fundamental types of streaming media: (i) material that originates from a source having a real-time nature, such as a radio or TV broadcast, and (ii) material that originates from a non-real-time source such as from a disk file. An example of non-real-time material might be a piece of music stored as a disk file, or a portion of a broadcast that originally was real-time, perhaps yesterday's TV evening news, and was recorded into a disk file. For purposes of clarity within this document, streaming media of type (i) will be referred to as "broadcast" media, and streaming media of type (ii) will be referred to as "file based" media.

Both streaming media types are handled similarly in conventional systems, and both are handled similarly by the streaming media buffering system of the present invention. The two streaming media types are readily distinguished. Broadcast streaming media has as its source a system or arrangement that by definition can only be transmitted to users as fast as the material is generated; for example, a disk jockey speaking into a microphone. File based media, on the other hand, can be transmitted to users at any data rate, since there is no inherent time element to a file residing on a computer disk. With conventional Internet streaming media systems for streaming media of either type, media data is transmitted from the server to the user at the rate at which it will be played out, regardless of the data rate capabilities of the connection between the server and the user.

Conventional streaming media systems may incorporate buffering systems for programmatic purposes. For example, the system may buffer media data at the server for the purpose of packet assembly/disassembly. Media data may also be buffered at the server to permit programming conveniences such as dealing with chunks of data of a specific size. Such server buffering of media data is not used by conventional streaming media systems to mitigate long term Internet performance degradation as described hereinafter.

The sending of audio or video files via a network is known in the art. U.S. Pat. No. 6,029,194 to Tilt describes a media server for the distribution of audio/video over networks, in which retrieved media frames are transferred to a FIFO buffer. A clock rate for a local clock is adjusted according to the fullness of the buffer. The media frames from the buffer are sent in the form of data packets over the networks in response to interrupts generated by the local clock. In this manner, the timing for the media frames is controlled by the user to assure a continuous stream of video during editing. U.S. Pat. No. 6,014,706 to Cannon, et al. discloses an apparatus and method for displaying streamed digital video data on a client computer. The client computer is configured to receive the streamed digital video data from a server computer via a computer network. The streamed digital video data is transmitted from the server computer to the client computer as a stream of video frames. U.S. Pat. No. 6,002,720, to Yurt, et al. discloses a system of distributing video and/or audio information wherein digital signal processing is employed to achieve high rates of data compression. U.S. Pat. No. 5,923,655, to Veschi et al. discloses a system and method for communicating audio/video data in a packet-based computer network wherein transmission of data packets through the computer network requires variable periods of transmission time. U.S. Pat. No. 5,922,048 to Emura discloses a video server apparatus having a stream control section which determines a keyframe readout interval and a keyframe playback interval that satisfy a playback speed designated by a terminal apparatus. Finally, U.S. Pat. No. 6,014,694 to Aharoni, et al. discloses a system and method for adaptively transporting video over networks, including the Internet, wherein the available bandwidth varies with time.

There remains a need in the art for a method and system that afford immediate and uninterrupted listening/viewing of streaming media by the user.

SUMMARY OF THE INVENTION

The present invention provides a system and method for sending streaming media, such as audio or video files, via the Internet. Immediate playing of the media on a user's computer is afforded while reducing interruptions in playback due to Internet congestion and temporary modem delays due to noisy lines. Nearly instantaneous playback is achieved, while maintaining protection against playback interruption. Delayed starts, heretofore required to provide protection against interruption, are avoided. Data loss due to interruptions in the receipt of media data by the media player can be recovered while the player continues to play out the audio or video material. If the interruptions are so severe as to deplete the user's buffer and stop the play out, the media player will begin to play out again as soon as the media player begins to receive media data without waiting to first build up the buffer.

Generally stated, the invention provides a system for distributing via the Internet streaming media composed of a plurality of time-sequenced data elements. The system has a server connected to the Internet for transmitting the data elements. Associated with the server are a buffer manager and a FIFO buffer for storing at least one of the data elements for transmission. The buffer manager comprises means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the FIFO buffer with a predetermined number of data elements; maintaining a pointer into the buffer for each user computer indicating the last media data element that has been sent to that user, thus indicating the next element or elements to be sent; and, once the FIFO buffer is full, deleting the oldest data elements in the buffer as new data

elements are received, said means arranged to maintain the pre-determined number of data elements in the FIFO buffer. At least one user computer is connected to the server via the Internet or other data communications medium.

This invention presumes the existence of a data communications transport mechanism, such as the TCP protocol, for the reliable delivery of data in an ordered sequence from the source of the media data to the server, or from the server to the media player software of the user computer. Thus, the delivery of data in the proper sequence is outside the scope of this invention.

The user computer is associated with a media player software incorporating a user buffer and comprises means for receiving and storing a predetermined number of media data elements which are received sequentially by the media player, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out. As data is played out, the next sequential data elements are received from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user's buffer.

There are two types of encoding schemes used for audio and video material—"Variable Bit Rate"—VBR, and "Constant Bit Rate"—CBR. CBR encoding represents the encoded media with a constant bit rate per second, regardless of the complexity of the material being encoded, for example, if an audio source is encoded at 20 kilobits per second at a Constant Bit Rate, the media data being produced from the encoding is at 20 kilobits per second whether the audio material is symphonic music or silence. Variable Bit Rate encoding uses a variable number of bits to represent sounds, with more bits required for complex (symphonic) sounds than for simple sounds or silence. The standard encoding scheme used for streaming media is CBR because the resulting data rate is more predictable than for VBR.

The server stores a predetermined amount of media data in a First-In First-Out (FIFO) buffer in an arrangement that receives media data directly or indirectly from a real-time source, such as a radio station. For example, the server buffer might be set to store up 30 seconds of media data. Because the source produces media data in real time, the media data is delivered to the server approximately at the rate it is generated. Of course there can be variability's in this data delivery process due to networking, disk accesses, and so on, causing the delivery rate of the media data to be variable over short periods of time, typically measured in seconds. But over a longer period of time measured in minutes or tens of minutes or longer, the media data is delivered from source to server at the rate it is generated, and the server in turn provides that media data to the FIFO buffer at that same rate. Since CBR encoding is normally used for streaming media, the media data is generated, received by the server, and provided to the buffer approximately at a fixed rate. Once the buffer is full, for each new data element received into the buffer the oldest data element is deleted from the buffer. Once a connection is made to a user's computer, the server sends the media data to the user computer's buffer in the following manner. First, media data is sent to the user at the highest rate that the data connection between the server and the user computer will support until the predetermined amount of data that had been stored in the server buffer has been transferred to the user's computer. Once the buffer has been transferred a steady state condition is reached wherein as each media data element arrives at the server, it is immediately sent out to the user computer. In this steady state condition, the media data is sent at a rate that matches the constant fill rate of the server buffer, and is received at the same rate by the user computer if there are no

interruptions in the transmission of media data between the server and the user's computer. If interruptions have interfered with the arrival of sent media data to the user's computer, that data may have been "dropped" by routers in the Internet and needs to be resent. This causes data to "back up" into the server FIFO for that user.

In one method of operation, the resending of missing data is the responsibility of a reliable transport mechanism, such as TCP. The server buffer "sends" data by delivering it to the transport mechanism. The transport mechanism actually "sends" the data across the communications medium, and has processes which determine if all the data that has been sent has been received by the destination. If not, missing pieces of data are automatically resent to the destination, and are arranged to be delivered to the target software on the destination system in an ordered fashion. In the circumstance of this invention, the destination is the user computer, and the target software on the destination system is the media player. If the transport mechanism determines that data is missing, it retransmits that data to the destination as fast as the connection between the server and destination will allow. The net effect of this invention is that all media data to be delivered to a user computer is always sent as fast as the communications medium will support, either by the server buffer passing media data to the transport mechanism, or by the transport mechanism delivering or redelivering the media data to the user computer. This is enabled by buffering data at the server, and is distinctly different from prior art in which media data is only sent from the server to the user computer at the rate at which it is to be played out.

In another method of operation, the server can use an unreliable transport mechanism, such as UDP, and rely on a streaming software process to manage data delivery and the resending of data elements not received by the media player.

As an example of the preceding description, if the server had been set to store 30 seconds of audio in its buffer, when a user connects that 30 seconds worth of media, data is transferred to the user's media player buffer as fast as the data connection between the two will allow. The media player can begin playing as soon as it has received a very minimum amount of data, perhaps comprising only a single packet of media data. For ease of understanding, consider the server buffer and the media player buffer to be an elastic system that between the two stores up 30 seconds of audio data. The server starts with 30 seconds of buffered audio data which it transfers to the media player until the server has no buffered media data and the media player has 30 seconds of buffered media data. Regardless of how much of the buffered media data has been transmitted to the media player, there always is 30 seconds of media data being buffered between the two locations. Consequently, the audio being played out by the media player will always be 30 seconds behind the audio at the source. If there were a media player in the radio station studio, an announcer would hear themselves through the media player with a 30 second delay.

Routinely, once a steady state has been achieved, the next data element to be sent is the next sequential data element from that which has already been received by the user's computer buffer. However, if there is more data to be sent than at the routine constant fill rate, such as in the condition where some media data has been resent by the reliable transport layer, the server transport mechanism will again send the buffered media data as fast as the connection between the server and the user's computer will support. Similarly, if the media player buffer begins to deplete or becomes depleted due to networking interruptions, the server will attempt to send as much data as is necessary to rebuild the user comput-

er's buffer to the proper level. This allows for rebuilding the user's computer buffer under circumstances wherein Internet interruptions have blocked the normal flow of data. When compared to conventional systems, which provide no capability to rebuild the user's computer buffer when data is lost, the streaming media buffering system of the present invention provides for recovery of lost data elements and the restoration of the user's buffer, even while the user media player continues to play.

Under conditions in which interruptions have interfered with the arrival of sent media data to the user's computer, data loss exceeding certain levels will cause the transport mechanism software to stop accepting data for transmission from the application software, namely the streaming media server software. Although other arrangements are possible within the scope of this invention, in the preferred embodiment, the streaming media server software keeps track of the last data element in the FIFO buffer that has been "sent" to each user using a software pointer. An interruption in the ability to send media data to a user results in this pointer "backing up" in the FIFO in such a way that the server knows from what point in the buffer to restart sending data when the transport mechanism again requests data to send. When the server software receives that notification, it will begin sending data to the user starting from the next data element to send as indicated by the pointer, and sending as much data as the transport mechanism will accept. The transport mechanism will again send this data as fast as it can to the user. This process continues until the steady state condition is again reached wherein each data element is sent to the user as soon as it arrives from the media source.

In another embodiment, the server is connected to the Internet, and to a broadcast media source, such as a radio station. A radio station computer is provided with a means for receiving media data elements as they are generated by the audio and/or video source, and for transmitting those media data elements to the server as they are generated. As before, the server provides a buffer manager and a FIFO buffer, and provides a means for receiving the sequentially arranged media data elements from the broadcast media source and storing those data elements in the FIFO buffer. The buffer manager comprises means for: supplying the FIFO buffer with a predetermined number of data elements; maintaining a pointer into the buffer for each user computer indicating the last media data element that has been sent to that user, thus indicating the next element or elements to be sent; and, once the FIFO buffer is full, deleting the oldest data element in the buffer as each new data element is received. Importantly, the buffer manager is arranged to maintain the pre-determined number of data elements in the FIFO buffer. At least one user computer is connected to the server via the Internet or other data communications medium.

The user computer is associated with a media player software incorporating a user buffer and comprises means for receiving and storing a predetermined number of media data elements, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out. As data is played out, the next sequential data elements are received from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user's buffer. It should be understood that data might arrive at the media player out-of-sequence and that processes in the media player or the media player buffer manager are responsible for properly arranging this data.

In another embodiment, the server is connected to the Internet and provisioned as initially described, and has available to it file based media data as the source material. The file based media data can be read by the server which can deliver media data elements to the server FIFO buffer at a constant time-sequenced rate, as if the data were arriving from a broadcast media source. As before, the server provides a buffer manager and a FIFO buffer, and provides a means for receiving the sequentially arranged media data elements from the file based media source and storing those data elements in the FIFO buffer. The buffer manager comprises means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the FIFO buffer with a predetermined number of data elements at a constant time-sequenced fill rate; maintaining a pointer into the buffer for each user computer indicating the last media data element that has been sent to that user, thus indicating the next element or elements to be sent; and, once the FIFO buffer is full, deleting the oldest data element in the buffer as each new data element is received, said means arranged to maintain the pre-determined number of data elements in the FIFO buffer. The server buffer manager, or a separate process on the server, or a process on another computer having access to the file based media data, provides for reading the media data file and making available to the FIFO buffer sequentially arranged media data elements. At least one user computer is connected to the server via the Internet.

The user computer is associated with a media player software incorporating a user buffer and comprises means for receiving and storing a predetermined number of media data elements, which are received sequentially by the media player, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out. As data is played out, the next sequential data elements are received from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user's buffer.

In another embodiment, the server is connected to the Internet and provisioned as initially described. The server buffer manager, or the media source, provides for sequentially numbering the media data elements. The server buffer manager does not maintain a pointer into the server buffer for each user. Instead, the media player buffer manager in the user computer maintains a record of the serial number of the last data element that has been received. Via the use of standard data communications protocol techniques such as TCP, the user computer transmits a request to the server to send one or more data elements, specifying the serial numbers of the data elements. The server responds by sending the requested data elements, and depends upon the reliable transmission protocol to assure delivery. The user computer then continues with additional data requests for the duration of playing the audio/video material. In this manner, the user computer, not the server, maintains the record of the highest data element number stored in the user computer buffer. The media data will be transmitted to the user computer as fast as the data connection between the user computer and the server will allow. As before, the server provides a buffer manager and a FIFO buffer, and provides a means for receiving the sequential media data elements from a broadcast media source or a file based media source, and storing those data elements in the FIFO buffer. The buffer manager comprises means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the FIFO buffer with a predetermined number of data elements; and, once the FIFO buffer is full, deleting the oldest data element in the buffer as each new data element is received. Such means is arranged to maintain the pre-determined number of data elements in the FIFO buffer. At least one user computer is connected to the server via the Internet.

The user computer is associated with a media player software incorporating a user buffer and comprises means for receiving and storing a predetermined number of media data elements, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out. As data is played out, the next sequential data elements are requested from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user's buffer. It should be understood that data might arrive at the media player out-of-sequence and that processes in the media player or the media player buffer manager are responsible for properly arranging this data.

In yet another embodiment, the invention provides a method for distributing from a server via the Internet streaming media composed of a plurality of time-sequenced data elements. A predetermined number of the data elements are sequentially loaded into a FIFO buffer. Additional data elements continue to be received. As each new data element is input to the buffer, the oldest data element is deleted from the buffer, maintaining in the buffer the same predetermined number of data elements. At the request of a user computer for connection to a media stream, a group of the data elements is sequentially sent via the Internet from the FIFO buffer to the user computer connected to the Internet. Upon being received by the user computer, the sent group of data elements is loaded into a user's buffer associated with the user computer. The users computer immediately begins to play the audio/video streaming media material. The server continues to send the next data elements in sequence until the contents of the FIFO buffer have been sent. The data elements are sent by the server as fast as the connection between the server and user computer will allow. Once the contents of the FIFO buffer have been sent to a user computer, as each new data element is received into the FIFO buffer it is immediately sent to the user computer in such a manner as to keep the user computer buffer full. The process repeats for substantially the entire time that the audio/video material is played.

Unlike conventional buffering systems, audio begins to play on the user system as soon as the user connection to the audio server is effected and a small amount of data has been transferred-conventional systems required many seconds of data. Audio/video media data is initially transmitted from the server more rapidly than it is played out by the user system, until the server buffer has been transferred to the user computer. The user's buffer is built up while the audio is playing, and can be restored if it is diminished by data transmission interruptions. Advantageously, the system and method of this invention afford faster data transmissions than the playback data rate of the media data. Audio/video data is transmitted from the server more rapidly than it is played out by the user system under conditions wherein the user's computer buffer is not full. The audio/video data in the user buffer accumulates; interruptions in playback due to temporary Internet and modem delays are avoided. It should be realized that, although the invention has been described hereinabove in connection with a process wherein the server sends buffered media data to the user "as fast as the network connection will permit", it is adequate, as mentioned in this paragraph, that the buffered data be transferred from the server to the user at a rate faster than the playback rate.

Although the preferred embodiment utilizes a reliable transport mechanism to move data between the server and the user, alternative embodiments could incorporate this invention's buffering system in combination with an unreliable datagram-based transport mechanism.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will be more fully understood and further advantages will become apparent when reference is had to the following detailed description and the accompanying drawings, in which:

FIG. **1** is a schematic diagram illustrating the elements of a streaming media buffering system in accordance with the present invention;

FIG. **2** is a schematic diagram of an alternative embodiment of the system shown by FIG. **1**; and

FIG. **3** is a block diagram illustrating the method of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention relates to a buffering system for streaming media, such as audio/video, on the Internet.

In one embodiment, the invention provides a system for distributing via the Internet streaming media composed of a plurality of time-sequenced data elements. As shown in FIG. **1**, the system is provided with a server **12** connected to the Internet **10** for transmitting the streaming media data elements. Associated with the server **12** is a FIFO buffer **14** for storing at least one of the data elements for transmission, and a buffer manager **16**. Buffer **14** is a conventional computer storage mechanism such as a hard disk, as shown for convenience of illustration, or, preferably, an electronic storage arrangement such as Random Access Memory (RAM). The buffer manager **16** is in the form of software or firmware that provides means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the buffer **14** with a predetermined number of data elements; maintaining a pointer **24***a* through **24***n* into the buffer, one for each user computer indicating the last media data element that has been sent to that user, thus indicating the next element or elements to be sent; and, once the FIFO buffer is full, deleting the oldest data element in the buffer as each new data element is received, said means arranged to maintain the pre-determined number of data elements in the FIFO buffer. Buffer Manager **16** may also comprise means for: formatting media data according to the requirements of buffer **14**, and for digitizing, encoding, and packetizing the media data.

There is at least one user computer **18** connected to the server **12** via the Internet **10**. A user buffer **20** is associated with the user computer **18**. The user buffer **20** is provided with means for storing a predetermined number of the data elements. User buffer **20** is a conventional computer storage mechanism such as a hard disk, or, preferably, an electronic storage arrangement such as Random Access Memory (RAM) as suggested by the illustration. A buffer manager **22** is associated with the user computer **18**. The buffer manager **22**, having the form of software or firmware, is provided with means for receiving and storing a predetermined number of media data elements which are received by the media player, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out.

The media may come from a live source, shown as **26** in FIG. **1**, or from a stored file on the server **12**, or another storage device, such as a hard drive. For broadcast media, as the term is used herein, such as an announcer speaking into a microphone, or playing a CD, the media source **26** can only transmit audio/video data as fast as it is generated. If the media source is file based, such as a music clip stored as a disk file, and if that disk file is stored on the server or an associated

server computer, this connection could be considered to be near instantaneous. In this case, rather than audio/video data filling and depleting the buffer **14**, an amount of audio/video data equivalent to the desired buffer size is logically constituted as a FIFO buffer. Such a construct is commonly called a data window. The data window moves on a time-sequenced basis through the media data file, thus defining the contents of the buffer on a moment-by-moment basis and performing the equivalent functions to receiving a new data element and deleting the oldest data element.

The server **12** maintains a buffer of audio/video data comprising an amount adequate to bridge gaps typical of Internet and modem delays to the user. Preferably, this buffer holds enough data elements for about one minute of play. The server buffer **14** is filled the first time the media source connection is established or a disk file is read.

Connections from the server **12** through the Internet **10** commonly are much faster than the data rate required for audio or video playback. This fact is insignificant for conventional servers because, not having a FIFO buffer or a buffer pointer for each user, audio/video data can only be sent as fast as it becomes available, or as fast as the pace at which it must be delivered to the user in order to be properly replayed. The user, typically interacting with "media player" software on their computer, selects an audio source requiring a data rate slower than that available by the user's connection to the Internet. For example, if the user's connection to the Internet is made via a 56,000 bits per second modem, the user might select a media source encoded for playback at 24,000 bits per second.

With the present invention, as soon as a user connects to the server **12**, the server **12** transmits audio/video data as sequential data elements from its buffer **14** to the buffer **20** of the user as fast as the data connection will allow. Unlike the prior art, media begins to play on the user computer **18** as soon as the user connection is made to the audio server **12** and a minimal amount of data elements have been received and stored in the user's buffer **20**. The user's buffer **20** is built up while the media is playing. As each data element is played, it is deleted from the user's buffer **20**. Initially, the user buffer manager **22** requests the server **12** to send media data elements to start the playback stream, such as by selecting a radio station from a list. The server **12** responds by sending data elements to the user computer **18** as fast as it can, until the entire FIFO buffer **14** has been sent to the user computer. Upon receipt of the initial data elements, the user buffer manager **22** begins playback. Because this is a synchronous system with the source, server, and user computer operating by the same playback clock rate as determined by the encoding rate of the media, as each data element is played out and is deleted from the user buffer **20**, another data element has been deposited into the server buffer **14** and is available to be sent to the user computer. Server **12** sends the newly available data elements as fast as the data rate of the connection between server **12** and user computer **18** will allow.

Since the connection from the Internet to the user is faster than that required for media playback, audio/video data is transmitted from the server faster than it is played out by the user system, thus building up audio/video data in the user buffer. For example, if the user's connection to the Internet is at 56,000 bits per second, and the data rate encoded for the media to be played is 24,000 bits per second, the buffer level of the user buffer **20** will fill at the rate of 32,000 bits per second (56,000 bits per second receive rate, minus 24,000 bits per second playout depletion rate).

If, for example, the server buffer **14** held one minute of audio/video data, eventually the user buffer **20** will hold one

minute of audio/video data. The effect is that, over a brief period of time, the server buffer **14**, or a designated portion of it, is transferred to the user buffer **20**. The number of data elements in the server buffer **14** actually never changes, it always maintains one minute of audio/video data. However, for the particular user under discussion, a copy of all the data held in the buffer has been sent to the user. Since the user buffer **20** now holds one minute of audio/video data, it can play continuously despite data reception interruptions of less than a minute, and as soon as the interruption ceases the user buffer **20** can begin to rebuild, which will take place as fast as the connection between the user computer **18** and the server **12** will allow. The media player can continue to play out the audio/video material while the user buffer **20** rebuilds.

The predetermined buffer level in the user buffer **20** may be set at less than the predetermined buffer level of the server buffer **14** if desired. For example, the server buffer **14** might be set to hold one minute of media data, and the user buffer **20** might be set to hold thirty seconds of media data.

Alternatively, the user computer is replaced by an Internet radio or Internet Appliance, which is comprised of a dedicated processor for receiving Internet radio or audio/video material. Examples of such devices might range from familiar computing devices such as palmtops, PDAs (Portable Digital Assistants), and wireless phones, to devices that appear and operate similarly to conventional consumer electronic devices such as radios and televisions, but with the additional capability of Internet access.

In another embodiment, as shown in FIG. **2**, the media source may be remote from the server **12**, such as a computer system **28** in a radio station studio. This computer includes a source manager **30** which may be implemented in software or firmware. The source manager **30** comprises means for: formatting media data according to the requirements of server **12**, buffer **14**, and buffer manager **16**; and, for transmitting that media data to server **12**. Source manager **30** may further include means for digitizing, encoding, and packetizing the media data. Media data typically is generated in real time such as by a speaker talking into a microphone or by playing a CD. Generally, computer system **28** transmits media data to server **12** in real time as the media data is generated. Buffering of media data might occur at computer system **28** for convenience of programming, but such buffering is incidental to the operation of the end-to-end system being described. Computer system **28** connects via the Internet **10**, or other suitable data communications medium, to a server **12**, wherein server buffer manager **16** receives the media data for input into the FIFO buffer **14** as described previously. Server **12**, in turn, transmits the media data to one or more user computers **18**, also as previously described. Network connections between the source computer and the server may cause media data to be delayed in arrival at the server, causing the server FIFO buffer occasionally to be less than full. In this circumstance, the server buffer transfers the media data that it has available in its buffer to the media player, and when more data arrives from the source, the server sends it out to the media player.

In another embodiment, shown in FIG. **3**, the invention provides a method for distributing from a server via the Internet streaming media composed of a plurality of time-sequenced data elements. Time-sequenced data elements are generated or received **32**. Next, a predetermined number of the data elements are sequentially loaded **34** into a server buffer, which process of **32** and **34** continues indefinitely as long as there is media data available. Next, a group of the data elements is sequentially sent **36** via the Internet from the server buffer to a user computer connected to the Internet. Upon receipt by the user computer, the sent group of data

elements is loaded **38** into a user buffer associated with the user computer. The user computer immediately plays **40** the received portion of the media on the user computer. At **42**, if the user buffer is not full, then additional data elements are sent to the user computer **36**. And also at **42**, if the user buffer is full, the system waits until new media data is delivered to the server buffer **34**. This process is repeated until the entire media file is played at the user computer.

Unlike conventional buffer arrangements, audio begins to play on the user system as soon as the user connection is made to the audio server. The user's buffer is built up while the audio is playing. Advantageously, the system and method of this invention create a faster than real time connection. That is to say, audio/video data is transmitted from the server faster than it is played out by the user system, thus building up audio/video data in the user buffer.

Having thus described the invention in rather full detail, it will be understood that such detail need not be strictly adhered to, but that additional changes and modifications may suggest themselves to one skilled in the art, all falling within the scope of the invention as defined by the subjoined claims.

I claim:

1. A method for distributing streaming media via a data communications medium such as the Internet to at least one user system of at least one user, the streaming media comprising a plurality of sequential media data elements for a digitally encoded audio or video program, comprising

providing a server programmed to receive requests from the user system for media data elements corresponding to specified serial identifiers and to send media data elements to the user system responsive to said requests, at a rate more rapid than the rate at which said streaming media is played back by a user; and

providing a machine-readable medium accessible to said user, on which there has been recorded software for implementing a media player for receiving and playing the streaming media on said user system, said software being programmed to cause the media player to maintain a record of the identifier of the last data element that has been received; and to transmit requests to the server to send one or more data elements, specifying the identifiers of the data elements, as said media player requires in order to maintain a sufficient number of media data elements in the media player for uninterrupted playback.

2. The method of claim **1**, wherein said serial identifiers are sequential.

3. The method of claim **1**, wherein said media is encoded at a constant bit rate.

4. The method of claim **1**, wherein said media is encoded at a variable bit rate.

5. The method of claim **1**, further comprising distributing said streaming media to a plurality of users.

6. The method of claim **5**, wherein said server does not maintain a pointer into a buffer established within said server, for each said user.

7. The method of claim **1**, wherein said server is adapted to receive said requests from said user system via standard communications protocol techniques such as TCP.

8. The method of claim **1**, wherein said server obtains said streaming media from a live source.

9. The method of claim **1**, wherein said server obtains said streaming media from a disk file.

10. A server for distributing streaming media via a data communications medium such as the Internet to at least one user system of at least one user, the streaming media comprising a plurality of sequential media data elements for a digitally encoded audio or video program, said user system being assumed to have a media player for receiving and playing the streaming media on said user system, which is operable to obtain media data elements from said server by transmitting requests to said server to send one or more specified media data elements, said server comprising

at least one data storage device, memory for storing machine-readable executable routines and for providing a working memory area for routines executing on the server, a central processing unit for executing the machine-readable executable routines, an operating system, at least one connection to the communications medium, and a communications system providing a set of communications protocols for communicating through said at least one connection;

a machine-readable, executable routine containing instructions to cause the server to assign serial identifiers to the sequential media data elements comprising the program;

a machine-readable, executable routine containing instructions to cause the server to receive requests from the user system for one or more media data elements specifying the identifiers of the requested data elements; and

a machine-readable, executable routine containing instructions to cause the server to send media data elements to the user system responsive to said requests, at a rate more rapid than the rate at which said streaming media is played back by a user.

11. The server of claim **10**, wherein said serial identifiers are sequential.

12. The server of claim **10**, wherein said media is encoded at a constant bit rate.

13. The server of claim **10**, wherein said media is encoded at a variable bit rate.

14. The server of claim **10**, wherein said server is adapted to distribute said streaming media to a plurality of simultaneous users.

15. The server of claim **10**, wherein said server does not maintain a pointer into a buffer established within said server, for each said user.

16. The server of claim **10**, wherein said operating system further comprises a protocol to receive said requests from said user system via standard communications protocol techniques such as TCP.

17. The server of claim **10**, wherein said server is adapted to obtain said streaming media from a live source.

18. The server of claim **10**, wherein said server obtains said streaming media from a disk file.

19. A non-transitory machine-readable medium on which there has been recorded a computer program for use in operating a computer to prepare streaming media content for transmission by a server wherein said server responds to user requests for media data elements identified by a serial identifier, said program recorded on said non-transitory machine readable medium comprising a routine to store and serially identify sequential data elements comprising said streaming media content, in a format capable of being served to users by said server.

20. The non-transitory machine readable medium of claim **19**, wherein said streaming media content is obtained by said computer from a disk file.

21. The non-transitory machine-readable medium of claim **19**, wherein said streaming media content is obtained from a live source.

22. The non-transitory machine-readable medium of claim **19**, wherein said streaming media content is encoded at a constant bit rate.

23. The non-transitory machine-readable medium of claim **19**, wherein said streaming media content is encoded at a variable bit rate.

24. A non-transitory machine-readable medium on which there has been recorded a computer program for use in operating a media player for receiving and playing streaming media comprising a plurality of sequential media data elements, said elements being available on request by said player via a data communications medium such as the Internet, from a server assumed to be capable of sending streaming media elements at a rate more rapid than the rate at which said streaming media is played back by a user, each said data element having a serial identifier, said program recorded on said machine readable medium comprising:

a routine that maintains a record of the identifier of the last sequential media data element that has been received by said player;

a routine that requests transmission of the next sequential media data elements following said last sequential

media data element, as said media player requires in order to maintain a sufficient number of media data elements in the media player for uninterrupted playback.

25. The non-transitory machine-readable medium of claim **24**, wherein said streaming media content is encoded at a constant bit rate.

26. The non-transitory machine-readable medium of claim **24**, wherein said streaming media content is encoded at a variable bit rate.

27. The non-transitory machine-readable medium of claim **24**, further comprising routines to send said requests to said server via standard communications protocol techniques such as TCP.

28. The method of claim **1**, wherein said server assigns serial identifiers to the sequential media data elements comprising the program.

* * * * *

(12) **INTER PARTES REVIEW CERTIFICATE** (1949th)
United States Patent (10) **Number: US 8,122,141 K1**
Price (45) **Certificate Issued: Mar. 1, 2021**

(54) **STREAMING MEDIA BUFFERING SYSTEM**

(75) Inventor: **Harold Edward Price**

(73) Assignee: **WAG Acquisition, LLC**

Trial Numbers:

IPR2016-01238 filed Jun. 21, 2016
IPR2017-00786 filed Jan. 27, 2017
IPR2017-00820 filed Jan. 31, 2017

Inter Partes Review Certificate for:

Patent No.: **8,122,141**
Issued: **Feb. 21, 2012**
Appl. No.: **12/800,152**
Filed: **May 10, 2010**

The results of IPR2016-01238 joined with IPR2017-00786 and IPR2017-00820 are reflected in this inter partes review certificate under 35 U.S.C. 318(b).

INTER PARTES REVIEW CERTIFICATE
U.S. Patent 8,122,141 K1
Trial No. IPR2016-01238
Certificate Issued Mar. 1, 2021

1 2

AS A RESULT OF THE INTER PARTES REVIEW PROCEEDING, IT HAS BEEN DETERMINED THAT:

Claims **10-23** are cancelled.

* * * * *

Exhibit 2
to
Ex Parte Reexamination
of
U.S. Patent No.
8,327,011

US008327011B2

(12) **United States Patent**
Price

(10) **Patent No.:** **US 8,327,011 B2**
(45) **Date of Patent:** **Dec. 4, 2012**

(54) **STREAMING MEDIA BUFFERING SYSTEM**

(75) Inventor: **Harold Edward Price**, Bethel Park, PA (US)

(73) Assignee: **WAG Acquistion, LLC**, Flanders, NJ (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **13/374,942**

(22) Filed: **Jan. 24, 2012**

(65) **Prior Publication Data**

US 2012/0151083 A1 Jun. 14, 2012

Related U.S. Application Data

(63) Continuation of application No. 12/800,152, filed on May 10, 2010, now Pat. No. 8,122,141, which is a continuation of application No. 10/893,814, filed on Jul. 19, 2004, now Pat. No. 7,716,358, which is a continuation-in-part of application No. 09/819,337, filed on Mar. 28, 2001, now Pat. No. 6,766,376.

(60) Provisional application No. 60/231,997, filed on Sep. 12, 2000.

(51) **Int. Cl.**
G06F 15/16 (2006.01)

(52) **U.S. Cl.** **709/231**

(58) **Field of Classification Search** 709/230, 709/231

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,963,995	A	10/1990	Lang
5,057,932	A	10/1991	Lang
5,065,396	A	11/1991	Castellano
5,164,839	A	11/1992	Lang
5,262,875	A	11/1993	Mincer et al.
5,361,259	A	11/1994	Hunt
5,414,455	A	5/1995	Hooper
5,440,334	A	8/1995	Walters et al.
5,481,543	A	1/1996	Veltman
5,493,514	A	2/1996	Keith
5,526,353	A	6/1996	Henley et al.
5,550,982	A	8/1996	Long
5,579,239	A	11/1996	Freeman
5,583,859	A	12/1996	Feldmeier
5,610,841	A	3/1997	Tanaka
5,613,032	A	3/1997	Cruz
5,627,936	A	5/1997	Prasad
5,663,951	A	9/1997	Danneels

(Continued)

FOREIGN PATENT DOCUMENTS

CA	2247588	C	1/2004

(Continued)

OTHER PUBLICATIONS

Rexford et al., "Smoothing variable-bit-rate video in an internetwork," IEEE/ACM Transactions on Networking, vol. 7, Issue 2, pp. 202-215, Apr. 1999.

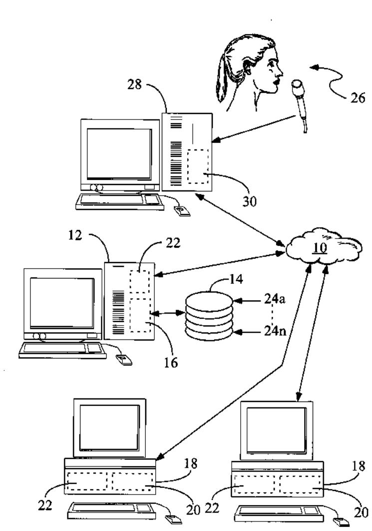
(Continued)

Primary Examiner — Larry Donaghue
Assistant Examiner — Marshall McLeod
(74) *Attorney, Agent, or Firm* — Ernest D. Buff; Ernest D. Buff Assoc. LLC

(57) **ABSTRACT**

Streaming media, such as audio or video files, is sent via the Internet. The media are immediately played on a user's computer. Audio/video data is transmitted from the server more rapidly than it is played out by the user system. The audio/video data in the user buffer accumulates; and interruptions in playback as well as temporary modem delays are avoided.

4 Claims, 3 Drawing Sheets



U.S. PATENT DOCUMENTS

5,668,948	A	9/1997	Belknap
5,710,970	A	1/1998	Walters et al.
5,719,786	A	2/1998	Nelson
5,734,119	A	3/1998	France
5,737,536	A	4/1998	Herrmann
5,751,951	A	5/1998	Osborne
5,751,968	A	5/1998	Cohen
5,768,527	A	6/1998	Zhu
5,778,374	A	7/1998	Dang
5,793,980	A	8/1998	Glaser et al.
5,805,823	A	9/1998	Seitz
5,809,239	A	9/1998	Dan
5,815,662	A	9/1998	Ong
5,819,160	A	10/1998	Foladare
5,821,986	A	10/1998	Yuan
5,822,537	A	10/1998	Katseff et al.
5,835,495	A	11/1998	Ferriere
5,867,230	A	2/1999	Wang
5,881,245	A	3/1999	Thompson
5,892,915	A	4/1999	Duso
5,910,876	A	6/1999	Sharma et al.
5,922,048	A	7/1999	Emura
5,923,655	A	7/1999	Veschi et al.
5,928,330	A	7/1999	Goetz
5,933,603	A	8/1999	Vahalia
5,938,734	A	8/1999	Yao
5,956,716	A	9/1999	Kenner
5,963,202	A	10/1999	Polish
5,974,503	A	10/1999	Venkatesh
5,978,567	A	11/1999	Rebane et al.
5,987,510	A	11/1999	Imai
5,995,705	A	11/1999	Lang
5,996,015	A	11/1999	Day
5,999,525	A	12/1999	Krishnaswamy et al.
6,002,720	A	12/1999	Yurt et al.
6,014,693	A	1/2000	Ito et al.
6,014,694	A	1/2000	Aharoni et al.
6,014,706	A	1/2000	Cannon et al.
6,029,194	A	2/2000	Tilt
6,032,193	A	2/2000	Sullivan
6,047,317	A	4/2000	Bisdikian et al.
6,047,356	A	4/2000	Anderson
6,057,832	A	5/2000	Lev et al.
6,061,731	A	5/2000	Blakeslee
6,061,732	A	5/2000	Korst et al.
6,065,050	A	5/2000	DeMoney
6,085,221	A	7/2000	Graf
6,151,632	A	11/2000	Chaddha
6,173,328	B1	1/2001	Sato
6,173,340	B1	1/2001	Gready et al.
6,233,226	B1	5/2001	Gringeri et al.
6,279,040	B1	8/2001	Ma et al.
6,292,834	B1	9/2001	Ravi et al.
6,321,269	B1	11/2001	Walker
6,377,995	B2	4/2002	Agraharam
6,385,596	B1	5/2002	Wiser
6,389,473	B1	5/2002	Carmel
6,405,256	B1	6/2002	Lin
6,430,620	B1	8/2002	Omura
6,449,719	B1	9/2002	Baker
6,452,943	B1	9/2002	Furuya
6,535,920	B1	3/2003	Parry
6,588,015	B1	7/2003	Eyer et al.
6,594,699	B1	7/2003	Sahai et al.
6,598,228	B2	7/2003	Hejna
6,625,750	B1	9/2003	Duso
6,637,031	B1	10/2003	Chou
6,665,751	B1	12/2003	Chen et al.
6,708,213	B1	3/2004	Bommaiah et al.
6,728,753	B1	4/2004	Parasnis
6,757,796	B1	6/2004	Hofmann
6,829,368	B2	12/2004	Meyer et al.
6,831,892	B2	12/2004	Robinett et al.
6,845,398	B1	1/2005	Galensky
6,850,965	B2	2/2005	Allen
6,889,257	B1	5/2005	Patel
6,907,481	B2	6/2005	Kovacevic
6,925,495	B2	8/2005	Hegde
6,938,047	B2	8/2005	Kryeziu
6,988,144	B1	1/2006	Luken
6,990,497	B2	1/2006	O'Rourke
7,020,710	B2	3/2006	Weber
7,039,784	B1	5/2006	Chen
7,054,500	B1	5/2006	Lillevoid
7,085,842	B2	8/2006	Reid
7,111,058	B1	9/2006	Nguyen et al.
7,113,983	B1	9/2006	Terada
7,170,856	B1	1/2007	Ho et al.
7,212,574	B2	5/2007	Abrams, Jr.
7,260,564	B1	8/2007	Lynn
7,272,298	B1	9/2007	Lang et al.
7,287,083	B1	10/2007	Nay
7,318,017	B2	1/2008	Swoboda
7,334,016	B2	2/2008	Fishhaut
7,334,044	B1	2/2008	Allen
7,346,698	B2	3/2008	Hannaway
7,349,663	B1	3/2008	Joseph
7,373,413	B1	5/2008	Nguyen et al.
7,376,710	B1	5/2008	Cromwell
7,471,834	B2	12/2008	Sull
7,478,164	B1	1/2009	Lango
7,496,676	B2	2/2009	Kryeziu
7,590,656	B2	9/2009	Plastina
7,647,297	B2	1/2010	LaChapelle
7,689,510	B2	3/2010	Lamkin
7,818,444	B2	10/2010	Brueck
7,890,631	B2	2/2011	Allen
7,917,557	B2	3/2011	Shteyn
2001/0047377	A1	11/2001	Sincaglia
2002/0007418	A1	1/2002	Hegde
2002/0023165	A1	2/2002	Lahr
2002/0029166	A1	3/2002	Jacobs
2002/0069218	A1	6/2002	Sull
2002/0078174	A1	6/2002	Sim
2002/0083182	A1	6/2002	Alvarado
2002/0120675	A1	8/2002	Everett
2002/0131443	A1	9/2002	Robinett
2002/0147634	A1	10/2002	Jacoby
2002/0177914	A1	11/2002	Chase
2003/0014488	A1	1/2003	Dalal
2003/0018978	A1	1/2003	Singal
2003/0061305	A1	3/2003	Copley
2003/0068046	A1	4/2003	Lindqvist
2003/0093790	A1	5/2003	Logan
2003/0186645	A1	10/2003	Mori
2004/0078812	A1	4/2004	Calvert
2004/0086120	A1	5/2004	Akins
2004/0123725	A1	7/2004	Kim
2004/0131340	A1	7/2004	Antoun
2004/0162910	A1	8/2004	Kryeziu
2004/0186733	A1	9/2004	Loomis
2004/0231004	A1	11/2004	Seo
2004/0260835	A1	12/2004	Welk
2005/0005025	A1	1/2005	Harville
2005/0080876	A1	4/2005	Peiffer
2005/0108320	A1	5/2005	Lord
2005/0190915	A1	9/2005	Pare
2005/0203917	A1	9/2005	Freeberg
2005/0251832	A1	11/2005	Chiueh
2005/0262251	A1	11/2005	Klemets
2006/0095472	A1	5/2006	Krikorian
2006/0143667	A1	6/2006	Kurosawa
2006/0153537	A1	7/2006	Kaneko
2007/0005428	A1	1/2007	Jacobs
2007/0016865	A1	1/2007	Johnson
2007/0038728	A1	2/2007	Jacobs
2007/0079327	A1	4/2007	Khoo
2007/0088804	A1	4/2007	Qureshey
2007/0226365	A1	9/2007	Hildreth
2007/0233784	A1	10/2007	O'Rourke
2007/0274672	A1	11/2007	Itoi
2008/0059532	A1	3/2008	Kazmi
2008/0133701	A1	6/2008	Kazmi
2008/0195743	A1	8/2008	Brueck

FOREIGN PATENT DOCUMENTS

EP	614317	A2	9/1994
EP	680185	A2	11/1995
EP	720374	A1	7/1996
EP	762300	A2	3/1997
EP	827336	A2	3/1998
EP	859535	A2	8/1998
EP	984584	A1	3/2000
EP	1113642	A2	7/2001
EP	1427218	A2	6/2004
JP	10336626	A	12/1998
JP	11184780	A	7/1999
JP	2000151595	A	5/2000
JP	2000172599	A	6/2000
JP	2000228669	A	8/2000
WO	9712447	A2	4/1997
WO	9717775	A1	5/1997
WO	9730551	A1	8/1997
WO	9847733	A1	10/1998
WO	0138993	A1	5/2001
WO	02057943	A1	7/2002
WO	03023781	A1	3/2003
WO	2005004485	A1	1/2005

OTHER PUBLICATIONS

Bianchi, “Buffer sizing for high speed video information retrieval on ATM networks,” GLOBECOM -New York-, 1997, vol. 2, pp. 1057-1061.

Salehi et al., “Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing,” IEEE/ACM Transactions on Networking, vol. 6, Issue 4, pp. 397-410, Aug. 1998.

Mielke et al., A Multi-level Buffering and Feedback Scheme for Distributed Multimedia Presentation Systems, IEEE, 1998.

Nam et al, Adaptive Multimedia Stream Presentation in Mobile Computing Environment, IEEE TENCON, Sep. 1999.

Deleon et al., An Adaptive Predictor for Media Playout Buffering, 1999.

Zhao et al., Bandwidth-Efficient Continuous Media Streaming through Optimal Multiplexing, 1999.

Gollapudi et al., Buffer Management in Multimedia Database Systems, 1996.

Hui et al., Client-Server Synchronization and Buffering for Variable Rate Multimedia Retrievals, IEEE Journal on Selected Areas in Communications, vol. 14, No. 1, Jan. 1996.

Zheng et al., Multimedia Over High Speed Networks: Reducing Network Requirements With Fast Buffer Fillup, 1998.

Zheng et al., Traffic Management of Multimedia over ATM Networks, Jan. 1999.

Chen et al., Video and Audio: Organization and Retrieval in the WWW, 1996.

Yee et al., Server and Buffer Allocation in Client Server Multimedia System, IEEE, 1993.

Rejaie et al., Rap: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet, IEEE, Mar. 1999.

Fig. 1

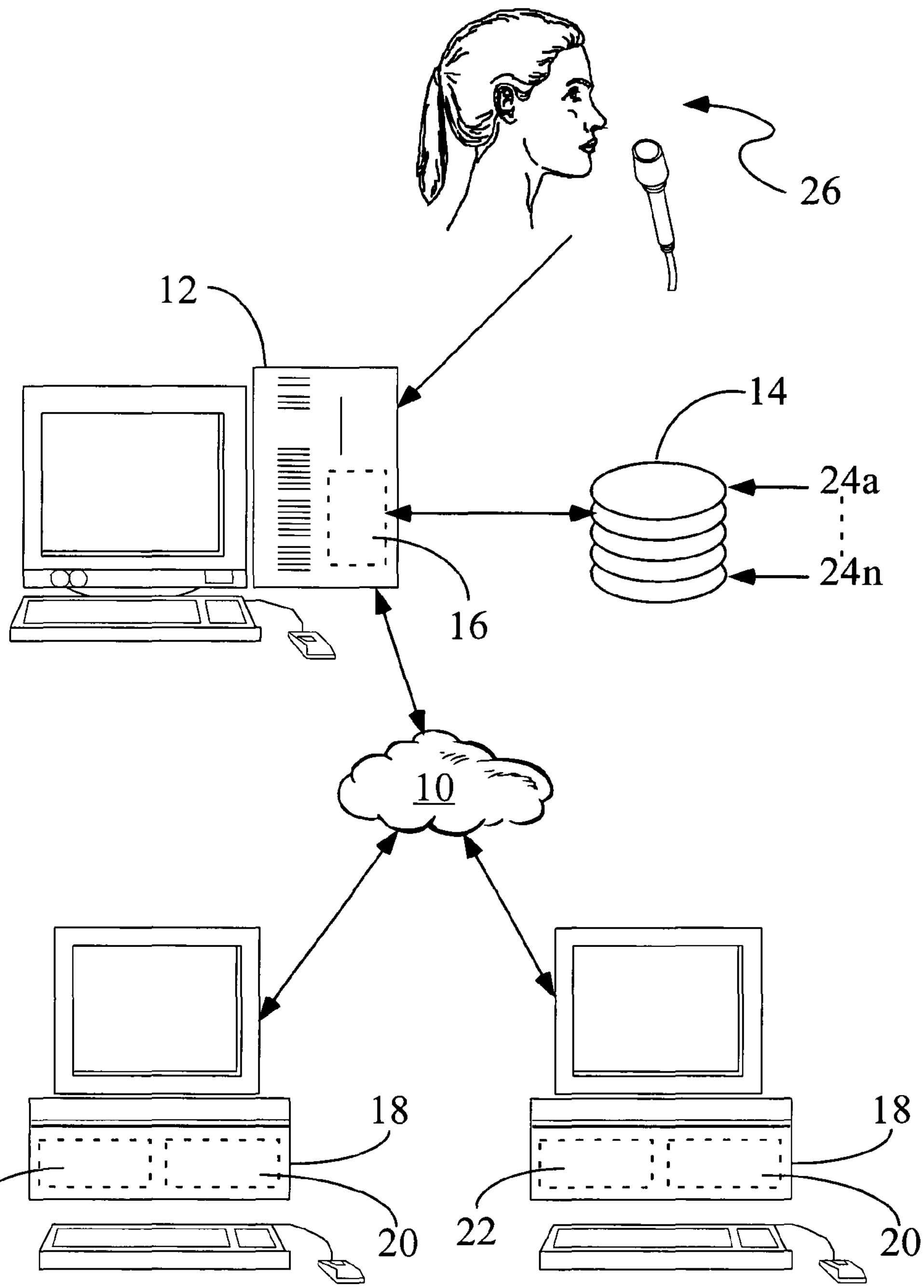


Fig. 2

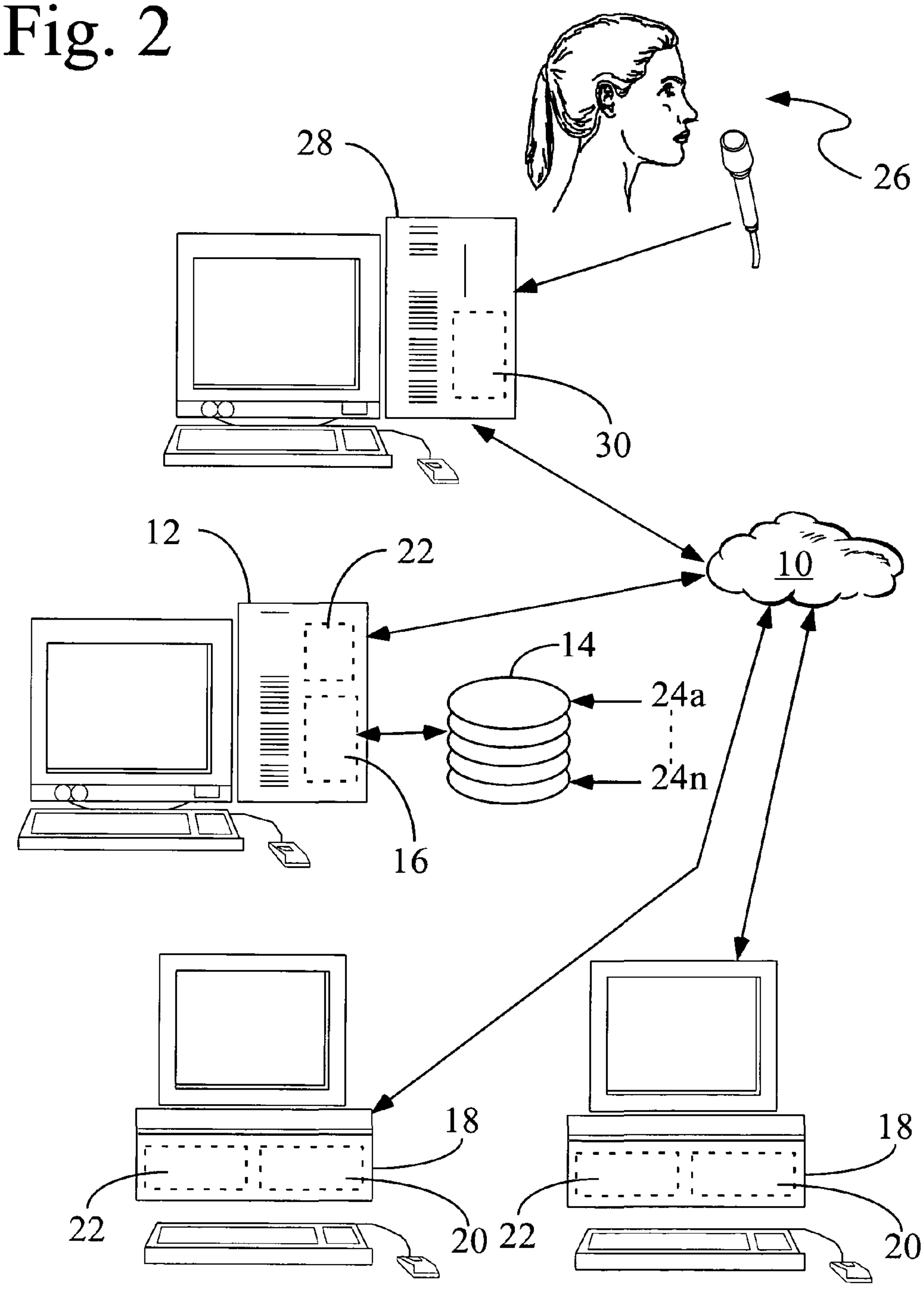
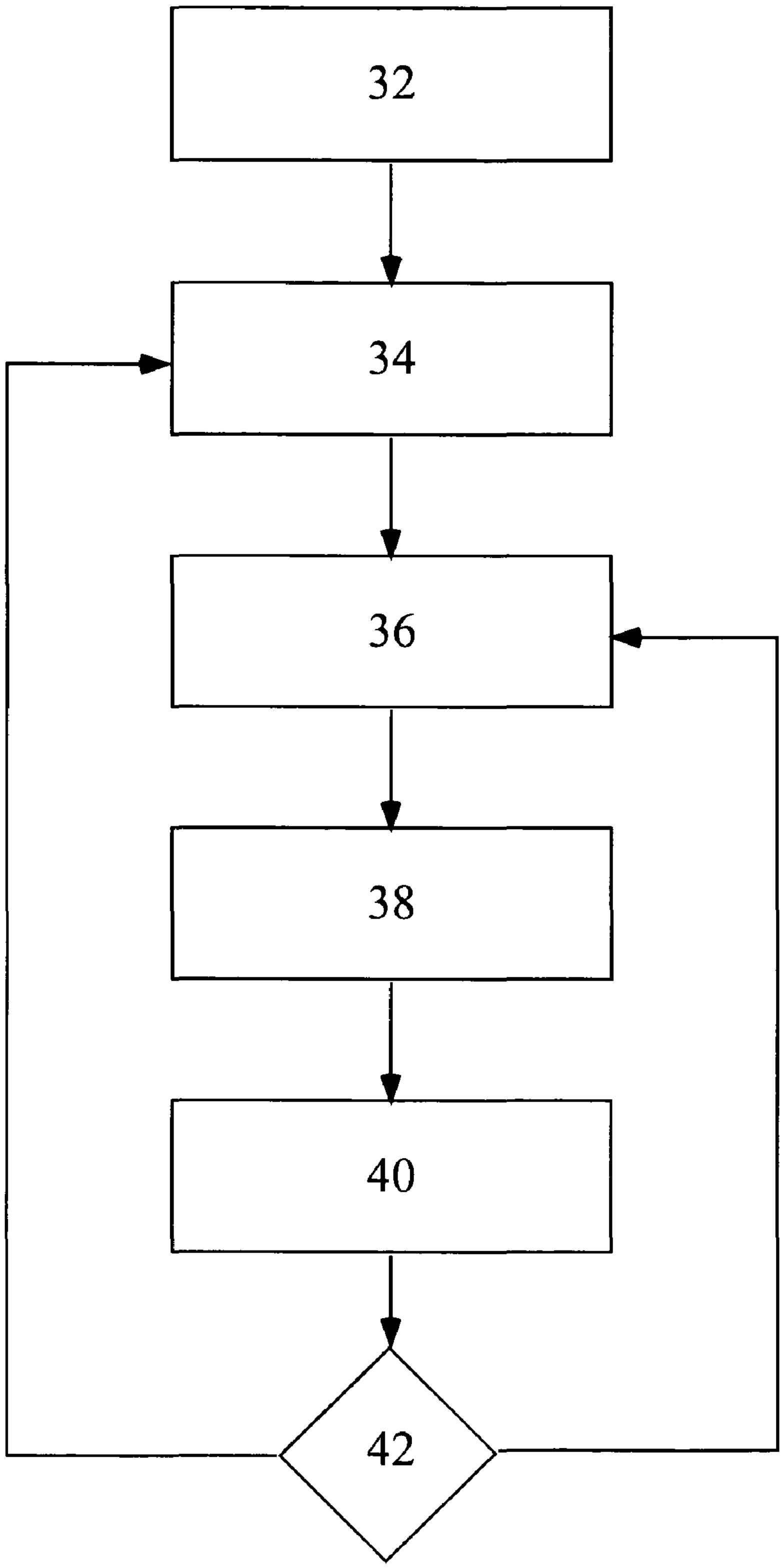


Fig. 3



STREAMING MEDIA BUFFERING SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 12/800,152, filed May 10, 2010 (published on Sep. 16, 2010 as U.S. patent publication number 2010/0235536 A1, and now U.S. Pat. No. 8,122,141), which was a continuation of U.S. patent application Ser. No. 10/893,814, filed Jul. 19, 2004 (published on Dec. 9, 2004 as U.S. patent publication number 2004/0249969 A1, and now U.S. Pat. No. 7,716,358), which was a continuation-in-part of U.S. patent application Ser. No. 09/819,337, filed Mar. 28, 2001 (now U.S. Pat. No. 6,766,376), which claimed the benefit under 35 U.S.C. §119(e) of U.S. provisional patent application Ser. No. 60/231,997, filed Sep. 12, 2000; and it claims the benefit, under 35 U.S.C. §120, of the respective filing dates of each of said applications, as well as benefit of the respective filing dates of U.S. patent application Ser. No. 12/800,177, filed May 10, 2010 (published on Sep. 2, 2010 as U.S. patent publication number 2010/0223362 A1, and now U.S. Pat. No. 8,185,611) which was also a continuation of said U.S. patent application Ser. No. 10/893,814, and of copending U.S. patent application Ser. No. 10/825,869, filed Apr. 16, 2004 (published on Dec. 23, 2004 as U.S. patent publication number 2004/260828 A1), which was a continuation of said U.S. patent application Ser. No. 09/819,337; and hereby incorporates by reference the entire disclosure of each of said prior applications.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to multimedia computer communication systems; and more particularly, to a buffering system for streaming media, such as audio/video, on the Internet.

2. Description of the Related Art

Prior to the development of Internet streaming media technologies, audio and video were formatted into files, which users needed to download to their computer before the files could be heard or viewed. Real time, continuous media, as from a radio station, was not suitable for this arrangement in that a file of finite size must be created so it could be downloaded. The advent of streaming media technologies allowed users to listen or view the files as they were being downloaded, and allowed users to "tune-in" to a continuous media broadcast, or "stream", such as from a radio station.

Since audio and video media must play out over a period of time it is more appropriate to think of bandwidth requirements than file size. The bandwidth requirement of an audio or video media refers to the data rate in bits per second that must be transmitted and received in order to listen or view the material uninterrupted. Transmitting the audio or video material over a connection slower than the bandwidth requirement results in unsatisfactory viewing or listening, if viewing or listening is possible at all. The connection available to most Internet users is by dial-up modem, which has a maximum receive data rate of 56,000 bits per second. Most audio and video available on the Internet has been compressed to be listenable or viewable within the 56,000 bits per second modem bandwidth. Requirements for achieving adequate audio and video over the Internet generally consume a considerable portion of the listener's available bandwidth.

Internet connection quality can vary rapidly over time, with two primary factors responsible for degradation of the instantaneous bandwidth actually available to the user. These factors are the quality of the user's modem connection over telephone lines, which can have periods of interference causing reduced available bandwidth, and momentary Internet congestion at various points along the route over which the user's data flows. Each of these factors can cause delays and interruptions in the transmission of data to the user. Internet data communications devices such as routers are designed to drop data "packets" if they get overloaded. For material that is not time sensitive, these dropped packets will usually be resent, and the user will eventually be presented with the material. However, since streaming media is time sensitive, dropped packets can have a significant impact on the receipt and playback of an audio or video stream. These degradations in the receipt of Internet data are very common, and prevent most users from being able to listen to or view streaming media without interruption unless some special provisions have been incorporated into the user's computer software to accommodate data transmission interruptions.

These interruptions are commonly referred to as "dropouts", meaning that the data flow to the user has been interrupted (i.e., the audio "drops out"). Dropouts can be extremely annoying—for example, while listening to music. The current state-of-the-art solution to the problem uses a pre-buffering technique to store up enough audio or video data in the user's computer so that it can play the audio or video with a minimum of dropouts. This process requires the user to wait until enough of the media file is buffered in memory before listening or viewing can begin. The media data is delivered by a server computer which has available to it the source of the media data, such as by a connection to a radio station. When the user connects to the server via the Internet, audio/video output at the user's system is delayed while the user's buffer is filled to a predetermined level. Typical pre-buffering wait times range from 10 to 20 seconds or more, determined by the vendor providing the audio or video media. Even with this pre-buffering process, interruptions in playback still occur.

In this process, the user has a software application on the computer commonly called a "media player". Using the features built into the media player, the user starts the audio or video stream, typically by clicking on a "start" button, and waits 10-20 seconds or so before the material starts playing. During this time data is being received from the source and filling the media player's buffer. The audio or video data is delivered from the source at the rate it is to be played out. If, for example, the user is listening to an audio stream encoded to be played-out at 24,000 bits per second, the source sends the audio data at the rate of 24,000 bits per second. Provided that the user waits 10 seconds, and the receipt of the buffering data has not been interrupted, there is enough media data stored in the buffer to play for 10 seconds.

Gaps in the receipt of audio/video data, due to Internet slowdowns, cause the buffer to deplete. Because transmission of audio/video media data to the user takes place at the rate it is played out, the user's buffer level can never be increased or replenished while it is playing. Thus, gaps in the receipt of audio/video media data inexorably cause the buffer level to decrease from its initial level. In time, extended or repeated occurrences of these gaps empty the user's buffer. The audio/video material stops playing, and the buffer must be refilled to its original predetermined level before playing of the media resumes.

By way of illustration in a 10 second pre-buffering scenario, if the data reception stopped the instant that the media started playing, it would play for exactly 10 seconds. Once it starts playing, the media data plays out of the buffer as new

media data replenishes the buffer. The incoming data rate equals the rate at which the data is played out of the user's buffer, assuming the receipt of data across the Internet is unimpeded. If there are no interruptions in the receipt of the media data for the duration of the time the user listens to or watches the material, the buffer level remains constant and there will still be 10 seconds of data stored in the media player's buffer when the user stops the player. On the other hand, if the media player encounters interruptions totaling 6 seconds while playing the material, there would only be 4 seconds of media data remaining in the buffer when the user stopped it. If data reception interruptions at any time during the playing exceed 10 seconds, the user's media player buffer becomes exhausted. There is no media data to play, and the audio or video stops—a dropout has occurred. At this point a software mechanism in the media player stops attempting to play any more of the material, and starts the buffering process again. The media player remains silent until the buffer refills, at which time the media player will once again start playing the material.

There are two fundamental types of streaming media: (i) material that originates from a source having a real-time nature, such as a radio or TV broadcast, and (ii) material that originates from a non-real-time source such as from a disk file. An example of non-real-time material might be a piece of music stored as a disk file, or a portion of a broadcast that originally was real-time, perhaps yesterday's TV evening news, and was recorded into a disk file. For purposes of clarity within this document, streaming media of type (i) will be referred to as "broadcast" media, and streaming media of type (ii) will be referred to as "file based" media.

Both streaming media types are handled similarly in conventional systems, and both are handled similarly by the streaming media buffering system of the present invention. The two streaming media types are readily distinguished. Broadcast streaming media has as its source a system or arrangement that by definition can only be transmitted to users as fast as the material is generated; for example, a disk jockey speaking into a microphone. File based media, on the other hand, can be transmitted to users at any data rate, since there is no inherent time element to a file residing on a computer disk. With conventional Internet streaming media systems for streaming media of either type, media data is transmitted from the server to the user at the rate at which it will be played out, regardless of the data rate capabilities of the connection between the server and the user.

Conventional streaming media systems may incorporate buffering systems for programmatic purposes. For example, the system may buffer media data at the server for the purpose of packet assembly/disassembly. Media data may also be buffered at the server to permit programming conveniences such as dealing with chunks of data of a specific size. Such server buffering of media data is not used by conventional streaming media systems to mitigate long term Internet performance degradation as described hereinafter.

The sending of audio or video files via a network is known in the art. U.S. Pat. No. 6,029,194 to Tilt describes a media server for the distribution of audio/video over networks, in which retrieved media frames are transferred to a FIFO buffer. A clock rate for a local clock is adjusted according to the fullness of the buffer. The media frames from the buffer are sent in the form of data packets over the networks in response to interrupts generated by the local clock. In this manner, the timing for the media frames is controlled by the user to assure a continuous stream of video during editing. U.S. Pat. No. 6,014,706 to Cannon, et al. discloses an apparatus and method for displaying streamed digital video data on a client computer. The client computer is configured to receive the streamed digital video data from a server computer via a computer network. The streamed digital video data is transmitted from the server computer to the client computer as a stream of video frames. U.S. Pat. No. 6,002,720, to Yurt, et al. discloses a system of distributing video and/or audio information wherein digital signal processing is employed to achieve high rates of data compression. U.S. Pat. No. 5,923,655, to Veschi et al. discloses a system and method for communicating audio/video data in a packet-based computer network wherein transmission of data packets through the computer network requires variable periods of transmission time. U.S. Pat. No. 5,922,048 to Emura discloses a video server apparatus having a stream control section which determines a keyframe readout interval and a keyframe playback interval that satisfy a playback speed designated by a terminal apparatus. Finally, U.S. Pat. No. 6,014,694 to Aharoni, et al. discloses a system and method for adaptively transporting video over networks, including the Internet, wherein the available bandwidth varies with time.

There remains a need in the art for a method and system that afford immediate and uninterrupted listening/viewing of streaming media by the user.

SUMMARY OF THE INVENTION

The present invention provides a system and method for sending streaming media, such as audio or video files, via the Internet. Immediate playing of the media on a user's computer is afforded while reducing interruptions in playback due to Internet congestion and temporary modem delays due to noisy lines. Nearly instantaneous playback is achieved, while maintaining protection against playback interruption. Delayed starts, heretofore required to provide protection against interruption, are avoided. Data loss due to interruptions in the receipt of media data by the media player can be recovered while the player continues to play out the audio or video material. If the interruptions are so severe as to deplete the user's buffer and stop the play out, the media player will begin to play out again as soon as the media player begins to receive media data without waiting to first build up the buffer.

Generally stated, the invention provides a system for distributing via the Internet streaming media composed of a plurality of time-sequenced data elements. The system has a server connected to the Internet for transmitting the data elements. Associated with the server are a buffer manager and a FIFO buffer for storing at least one of the data elements for transmission. The buffer manager comprises means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the FIFO buffer with a predetermined number of data elements; maintaining a pointer into the buffer for each user computer indicating the last media data element that has been sent to that user, thus indicating the next element or elements to be sent; and, once the FIFO buffer is full, deleting the oldest data elements in the buffer as new data elements are received, said means arranged to maintain the pre-determined number of data elements in the FIFO buffer. At least one user computer is connected to the server via the Internet or other data communications medium.

This invention presumes the existence of a data communications transport mechanism, such as the TCP protocol, for the reliable delivery of data in an ordered sequence from the source of the media data to the server, or from the server to the media player software of the user computer. Thus, the delivery of data in the proper sequence is outside the scope of this invention.

The user computer is associated with a media player software incorporating a user buffer and comprises means for receiving and storing a predetermined number of media data elements which are received sequentially by the media player, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out. As data is played out, the next sequential data elements are received from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user's buffer.

There are two types of encoding schemes used for audio and video material—"Variable Bit Rate"—VBR, and "Constant Bit Rate"—CBR. CBR encoding represents the encoded media with a constant bit rate per second, regardless of the complexity of the material being encoded, for example, if an audio source is encoded at 20 kilobits per second at a Constant Bit Rate, the media data being produced from the encoding is at 20 kilobits per second whether the audio material is symphonic music or silence. Variable Bit Rate encoding uses a variable number of bits to represent sounds, with more bits required for complex (symphonic) sounds than for simple sounds or silence. The standard encoding scheme used for streaming media is CBR because the resulting data rate is more predictable than for VBR.

The server stores a predetermined amount of media data in a First-In First-Out (FIFO) buffer in an arrangement that receives media data directly or indirectly from a real-time source, such as a radio station. For example, the server buffer might be set to store up 30 seconds of media data. Because the source produces media data in real time, the media data is delivered to the server approximately at the rate it is generated. Of course there can be variability's in this data delivery process due to networking, disk accesses, and so on, causing the delivery rate of the media data to be variable over short periods of time, typically measured in seconds. But over a longer period of time measured in minutes or tens of minutes or longer, the media data is delivered from source to server at the rate it is generated, and the server in turn provides that media data to the FIFO buffer at that same rate. Since CBR encoding is normally used for streaming media, the media data is generated, received by the server, and provided to the buffer approximately at a fixed rate. Once the buffer is full, for each new data element received into the buffer the oldest data element is deleted from the buffer. Once a connection is made to a user's computer, the server sends the media data to the user computer's buffer in the following manner. First, media data is sent to the user at the highest rate that the data connection between the server and the user computer will support until the predetermined amount of data that had been stored in the server buffer has been transferred to the user's computer. Once the buffer has been transferred a steady state condition is reached wherein as each media data element arrives at the server, it is immediately sent out to the user computer. In this steady state condition, the media data is sent at a rate that matches the fill rate of the server buffer, and is received at the same rate by the user computer if there are no interruptions in the transmission of media data between the server and the user's computer. If interruptions have interfered with the arrival of sent media data to the user's computer, that data may have been "dropped" by routers in the Internet and needs to be resent. This causes data to "back up" into the server FIFO for that user.

In one method of operation, the resending of missing data is the responsibility of a reliable transport mechanism, such as TCP. The server buffer "sends" data by delivering it to the transport mechanism. The transport mechanism actually "sends" the data across the communications medium, and has processes which determine if all the data that has been sent has been received by the destination. If not, missing pieces of data are automatically resent to the destination, and are arranged to be delivered to the target software on the destination system in an ordered fashion. In the circumstance of this invention, the destination is the user computer, and the target software on the destination system is the media player. If the transport mechanism determines that data is missing, it retransmits that data to the destination as fast as the connection between the server and destination will allow. The net effect of this invention is that all media data to be delivered to a user computer is always sent as fast as the communications medium will support, either by the server buffer passing media data to the transport mechanism, or by the transport mechanism delivering or redelivering the media data to the user computer. This is enabled by buffering data at the server, and is distinctly different from prior art in which media data is only sent from the server to the user computer at the rate at which it is to be played out.

In another method of operation, the server can use an unreliable transport mechanism, such as UDP, and rely on a streaming software process to manage data delivery and the resending of data elements not received by the media player.

As an example of the preceding description, if the server had been set to store 30 seconds of audio in its buffer, when a user connects that 30 seconds worth of media, data is transferred to the user's media player buffer as fast as the data connection between the two will allow. The media player can begin playing as soon as it has received a very minimum amount of data, perhaps comprising only a single packet of media data. For ease of understanding, consider the server buffer and the media player buffer to be an elastic system that between the two stores up 30 seconds of audio data. The server starts with 30 seconds of buffered audio data which it transfers to the media player until the server has no buffered media data and the media player has 30 seconds of buffered media data. Regardless of how much of the buffered media data has been transmitted to the media player, there always is 30 seconds of media data being buffered between the two locations. Consequently, the audio being played out by the media player will always be 30 seconds behind the audio at the source. If there were a media player in the radio station studio, an announcer would hear themselves through the media player with a 30 second delay.

Routinely, once a steady state has been achieved, the next data element to be sent is the next sequential data element from that which has already been received by the user's computer buffer. However, if there is more data to be sent than at the routine constant fill rate, such as in the condition where some media data has been resent by the reliable transport layer, the server transport mechanism will again send the buffered media data as fast as the connection between the server and the user's computer will support. Similarly, if the media player buffer begins to deplete or becomes depleted due to networking interruptions, the server will attempt to send as much data as is necessary to rebuild the user computer's buffer to the proper level. This allows for rebuilding the user's computer buffer under circumstances wherein Internet interruptions have blocked the normal flow of data. When compared to conventional systems, which provide no capability to rebuild the user's computer buffer when data is lost, the streaming media buffering system of the present invention provides for recovery of lost data elements and the restoration of the user's buffer, even while the user media player continues to play.

Under conditions in which interruptions have interfered with the arrival of sent media data to the user's computer, data

loss exceeding certain levels will cause the transport mechanism software to stop accepting data for transmission from the application software, namely the streaming media server software. Although other arrangements are possible within the scope of this invention, in the preferred embodiment, the streaming media server software keeps track of the last data element in the FIFO buffer that has been "sent" to each user using a software pointer. An interruption in the ability to send media data to a user results in this pointer "backing Up11 in the FIFO in such a way that the server knows from what point in the buffer to restart sending data when the transport mechanism again requests data to send. When the server software receives that notification, it will begin sending data to the user starting from the next data element to send as indicated by the pointer, and sending as much data as the transport mechanism will accept. The transport mechanism will again send this data as fast as it can to the user. This process continues until the steady state condition is again reached wherein each data element is sent to the user as soon as it arrives from the media source.

In another embodiment, the server is connected to the Internet, and to a broadcast media source, such as a radio station. A radio station computer is provided with a means for receiving media data elements as they are generated by the audio and/or video source, and for transmitting those media data elements to the server as they are generated. As before, the server provides a buffer manager and a FIFO buffer, and provides a means for receiving the sequentially arranged media data elements from the broadcast media source and storing those data elements in the FIFO buffer. The buffer manager comprises means for: supplying the FIFO buffer with a predetermined number of data elements; maintaining a pointer into the buffer for each user computer indicating the last media data element that has been sent to that user, thus indicating the next element or elements to be sent; and, once the FIFO buffer is full, deleting the oldest data element in the buffer as each new data element is received. Importantly, the buffer manager is arranged to maintain the pre-determined number of data elements in the FIFO buffer. At least one user computer is connected to the server via the Internet or other data communications medium.

The user computer is associated with a media player software incorporating a user buffer and comprises means for receiving and storing a predetermined number of media data elements, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out. As data is played out, the next sequential data elements are received from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user's buffer. It should be understood that data might arrive at the media player out-of-sequence and that processes in the media player or the media player buffer manager are responsible for properly arranging this data.

In another embodiment, the server is connected to the Internet and provisioned as initially described, and has available to it file based media data as the source material. The file based media data can be read by the server which can deliver media data elements to the server FIFO buffer at a constant time-sequenced rate, as if the data were arriving from a broadcast media source. As before, the server provides a buffer manager and a FIFO buffer, and provides a means for receiving the sequentially arranged media data elements from the file based media source and storing those data elements in the FIFO buffer. The buffer manager comprises means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the FIFO buffer with a predetermined number of data elements at a constant time-sequenced fill rate; maintaining a pointer into the buffer for each user computer indicating the last media data element that has been sent to that user, thus indicating the next element or elements to be sent; and, once the FIFO buffer is full, deleting the oldest data element in the buffer as each new data element is received, said means arranged to maintain the pre-determined number of data elements in the FIFO buffer. The server buffer manager, or a separate process on the server, or a process on another computer having access to the file based media data, provides for reading the media data file and making available to the FIFO buffer sequentially arranged media data elements. At least one user computer is connected to the server via the Internet.

The user computer is associated with a media player software incorporating a user buffer and comprises means for receiving and storing a predetermined number of media data elements, which are received sequentially by the media player, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out. As data is played out, the next sequential data elements are received from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user's buffer.

In another embodiment, the server is connected to the Internet and provisioned as initially described. The server buffer manager, or the media source, provides for sequentially numbering the media data elements. The server buffer manager does not maintain a pointer into the server buffer for each user. Instead, the media player buffer manager in the user computer maintains a record of the serial number of the last data element that has been received. Via the use of standard data communications protocol techniques such as TCP, the user computer transmits a request to the server to send one or more data elements, specifying the serial numbers of the data elements. The server responds by sending the requested data elements, and depends upon the reliable transmission protocol to assure delivery. The user computer then continues with additional data requests for the duration of playing the audio/video material. In this manner, the user computer, not the server, maintains the record of the highest data element number stored in the user computer buffer. The media data will be transmitted to the user computer as fast as the data connection between the user computer and the server will allow. As before, the server provides a buffer manager and a FIFO buffer, and provides a means for receiving the sequential media data elements from a broadcast media source or a file based media source, and storing those data elements in the FIFO buffer. The buffer manager comprises means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the FIFO buffer with a predetermined number of data elements; and, once the FIFO buffer is full, deleting the oldest data element in the buffer as each new data element is received. Such means is arranged to maintain the pre-determined number of data elements in the FIFO buffer. At least one user computer is connected to the server via the Internet.

The user computer is associated with a media player software incorporating a user buffer and comprises means for receiving and storing a predetermined number of media data elements, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out. As data is played out, the next sequential data elements are requested from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user's buffer. It should be understood that data might arrive at the media player out-of-sequence and that

processes in the media player or the media player buffer manager are responsible for properly arranging this data.

In yet another embodiment, the invention provides a method for distributing from a server via the Internet streaming media composed of a plurality of time-sequenced data elements. A predetermined number of the data elements are sequentially loaded into a FIFO buffer. Additional data elements continue to be received. As each new data element is input to the buffer, the oldest data element is deleted from the buffer, maintaining in the buffer the same predetermined number of data elements. At the request of a user computer for connection to a media stream, a group of the data elements is sequentially sent via the Internet from the FIFO buffer to the user computer connected to the Internet. Upon being received by the user computer, the sent group of data elements is loaded into a user's buffer associated with the user computer. The user's computer immediately begins to play the audio/video streaming media material. The server continues to send the next data elements in sequence until the contents of the FIFO buffer have been sent. The data elements are sent by the server as fast as the connection between the server and user computer will allow. Once the contents of the FIFO buffer have been sent to a user computer, as each new data element is received into the FIFO buffer it is immediately sent to the user computer in such a manner as to keep the user computer buffer full. The process repeats for substantially the entire time that the audio/video material is played.

Unlike conventional buffering systems, audio begins to play on the user system as soon as the user connection to the audio server is effected and a small amount of data has been transferred-conventional systems required many seconds of data. Audio/video media data is initially transmitted from the server more rapidly than it is played out by the user system, until the server buffer has been transferred to the user computer. The user's buffer is built up while the audio is playing, and can be restored if it is diminished by data transmission interruptions. Advantageously, the system and method of this invention afford faster data transmissions than the playback data rate of the media data. Audio/video data is transmitted from the server more rapidly than it is played out by the user system under conditions wherein the user's computer buffer is not full. The audio/video data in the user buffer accumulates; interruptions in playback due to temporary Internet and modem delays are avoided. It should be realized that, although the invention has been described hereinabove in connection with a process wherein the server sends buffered media data to the user "as fast as the network connection will permit", it is adequate, as mentioned in this paragraph, that the buffered data be transferred from the server to the user at a rate faster than the playback rate.

Although the preferred embodiment utilizes a reliable transport mechanism to move data between the server and the user, alternative embodiments could incorporate this invention's buffering system in combination with an unreliable datagram-based transport mechanism.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will be more fully understood and further advantages will become apparent when reference is had to the following detailed description and the accompanying drawings, in which:

FIG. **1** is a schematic diagram illustrating the elements of a streaming media buffering system in accordance with the present invention;

FIG. **2** is a schematic diagram of an alternative embodiment of the system shown by FIG. **1**; and

FIG. **3** is a block diagram illustrating the method of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention relates to a buffering system for streaming media, such as audio/video, on the Internet.

In one embodiment, the invention provides a system for distributing via the Internet streaming media composed of a plurality of time-sequenced data elements. As shown in FIG. **1**, the system is provided with a server **12** connected to the Internet **10** for transmitting the streaming media data elements. Associated with the server **12** is a FIFO buffer **14** for storing at least one of the data elements for transmission, and a buffer manager **16**. Buffer **14** is a conventional computer storage mechanism such as a hard disk, as shown for convenience of illustration, or, preferably, an electronic storage arrangement such as Random Access Memory (RAM). The buffer manager **16** is in the form of software or firmware that provides means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the buffer **14** with a predetermined number of data elements; maintaining a pointer **24***a* through **24***n* into the buffer, one for each user computer indicating the last media data element that has been sent to that user, thus indicating the next element or elements to be sent; and, once the FIFO buffer is full, deleting the oldest data element in the buffer as each new data element is received, said means arranged to maintain the pre-determined number of data elements in the FIFO buffer. Buffer Manager **16** may also comprise means for: formatting media data according to the requirements of buffer **14**, and for digitizing, encoding, and packetizing the media data.

There is at least one user computer **18** connected to the server **12** via the Internet **10**. A user buffer **20** is associated with the user computer **18**. The user buffer **20** is provided with means for storing a predetermined number of the data elements. User buffer **20** is a conventional computer storage mechanism such as a hard disk, or, preferably, an electronic storage arrangement such as Random Access Memory (RAM) as suggested by the illustration. A buffer manager **22** is associated with the user computer **18**. The buffer manager **22**, having the form of software or firmware, is provided with means for receiving and storing a predetermined number of media data elements which are received by the media player, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out.

The media may come from a live source, shown as **26** in FIG. **1**, or from a stored file on the server **12**, or another storage device, such as a hard drive. For broadcast media, as the term is used herein, such as an announcer speaking into a microphone, or playing a CD, the media source **26** can only transmit audio/video data as fast as it is generated. If the media source is file based, such as a music clip stored as a disk file, and if that disk file is stored on the server or an associated server computer, this connection could be considered to be near instantaneous. In this case, rather than audio/video data filling and depleting the buffer **14**, an amount of audio/video data equivalent to the desired buffer size is logically constituted as a FIFO buffer. Such a construct is commonly called a data window. The data window moves on a time-sequenced basis through the media data file, thus defining the contents of the buffer on a moment-by-moment basis and performing the equivalent functions to receiving a new data element and deleting the oldest data element.

The server **12** maintains a buffer of audio/video data comprising an amount adequate to bridge gaps typical of Internet and modem delays to the user. Preferably, this buffer holds enough data elements for about one minute of play. The server buffer **14** is filled the first time the media source connection is established or a disk file is read.

Connections from the server **12** through the Internet **10** commonly are much faster than the data rate required for audio or video playback. This fact is insignificant for conventional servers because, not having a FIFO buffer or a buffer pointer for each user, audio/video data can only be sent as fast as it becomes available, or as fast as the pace at which it must be delivered to the user in order to be properly replayed. The user, typically interacting with "media player" software on their computer, selects an audio source requiring a data rate slower than that available by the user's connection to the Internet. For example, if the user's connection to the Internet is made via a 56,000 bits per second modem, the user might select a media source encoded for playback at 24,000 bits per second.

With the present invention, as soon as a user connects to the server **12**, the server **12** transmits audio/video data as sequential data elements from its buffer **14** to the buffer **20** of the user as fast as the data connection will allow. Unlike the prior art, media begins to play on the user computer **18** as soon as the user connection is made to the audio server **12** and a minimal amount of data elements have been received and stored in the user's buffer **20**. The user's buffer **20** is built up while the media is playing. As each data element is played, it is deleted from the user's buffer **20**. Initially, the user buffer manager **22** requests the server **12** to send media data elements to start the playback stream, such as by selecting a radio station from a list. The server **12** responds by sending data elements to the user computer **18** as fast as it can, until the entire FIFO buffer **14** has been sent to the user computer. Upon receipt of the initial data elements, the user buffer manager **22** begins playback. Because this is a synchronous system with the source, server, and user computer operating by the same playback clock rate as determined by the encoding rate of the media, as each data element is played out and is deleted from the user buffer **20**, another data element has been deposited into the server buffer **14** and is available to be sent to the user computer. Server **12** sends the newly available data elements as fast as the data rate of the connection between server **12** and user computer **18** will allow.

Since the connection from the Internet to the user is faster than that required for media playback, audio/video data is transmitted from the server faster than it is played out by the user system, thus building up audio/video data in the user buffer. For example, if the user's connection to the Internet is at 56,000 bits per second, and the data rate encoded for the media to be played is 24,000 bits per second, the buffer level of the user buffer **20** will fill at the rate of 32,000 bits per second (56,000 bits per second receive rate, minus 24,000 bits per second playout depletion rate).

If, for example, the server buffer **14** held one minute of audio/video data, eventually the user buffer **20** will hold one minute of audio/video data. The effect is that, over a brief period of time, the server buffer **14**, or a designated portion of it, is transferred to the user buffer **20**. The number of data elements in the server buffer **14** actually never changes, it always maintains one minute of audio/video data. However, for the particular user under discussion, a copy of all the data held in the buffer has been sent to the user. Since the user buffer **20** now holds one minute of audio/video data, it can play continuously despite data reception interruptions of less than a minute, and as soon as the interruption ceases the user buffer **20** can begin to rebuild, which will take place as fast as the connection between the user computer **18** and the server **12** will allow. The media player can continue to play out the audio/video material while the user buffer **20** rebuilds.

The predetermined buffer level in the user buffer **20** may be set at less than the predetermined buffer level of the server buffer **14** if desired. For example, the server buffer **14** might be set to hold one minute of media data, and the user buffer **20** might be set to hold thirty seconds of media data.

Alternatively, the user computer is replaced by an Internet radio or Internet Appliance, which is comprised of a dedicated processor for receiving Internet radio or audio/video material. Examples of such devices might range from familiar computing devices such as palmtops, PDAs (Portable Digital Assistants), and wireless phones, to devices that appear and operate similarly to conventional consumer electronic devices such as radios and televisions, but with the additional capability of Internet access.

In another embodiment, as shown in FIG. **2**, the media source may be remote from the server **12**, such as a computer system **28** in a radio station studio. This computer includes a source manager **30** which may be implemented in software or firmware. The source manager **30** comprises means for: formatting media data according to the requirements of server **12**, buffer **14**, and buffer manager **16**; and, for transmitting that media data to server **12**. Source manager **30** may further include means for digitizing, encoding, and packetizing the media data. Media data typically is generated in real time such as by a speaker talking into a microphone or by playing a CD. Generally, computer system **28** transmits media data to server **12** in real time as the media data is generated. Buffering of media data might occur at computer system **28** for convenience of programming, but such buffering is incidental to the operation of the end-to-end system being described. Computer system **28** connects via the Internet **10**, or other suitable data communications medium, to a server **12**, wherein server buffer manager **16** receives the media data for input into the FIFO buffer **14** as described previously. Server **12**, in turn, transmits the media data to one or more user computers **18**, also as previously described. Network connections between the source computer and the server may cause media data to be delayed in arrival at the server, causing the server FIFO buffer occasionally to be less than full. In this circumstance, the server buffer transfers the media data that it has available in its buffer to the media player, and when more data arrives from the source, the server sends it out to the media player.

In another embodiment, shown in FIG. **3**, the invention provides a method for distributing from a server via the Internet streaming media composed of a plurality of time-sequenced data elements. Time-sequenced data elements are generated or received **32**. Next, a predetermined number of the data elements are sequentially loaded **34** into a server buffer, which process of **32** and **34** continues indefinitely as long as there is media data available. Next, a group of the data elements is sequentially sent **36** via the Internet from the server buffer to a user computer connected to the Internet. Upon receipt by the user computer, the sent group of data elements is loaded **38** into a user buffer associated with the user computer. The user computer immediately plays **40** the received portion of the media on the user computer. At **42**, if the user buffer is not full, then additional data elements are sent to the user computer **36**. And also at **42**, if the user buffer is full, the system waits until new media data is delivered to the server buffer **34**. This process is repeated until the entire media file is played at the user computer.

Unlike conventional buffer arrangements, audio begins to play on the user system as soon as the user connection is made

to the audio server. The user's buffer is built up while the audio is playing. Advantageously, the system and method of this invention create a faster than real time connection. That is to say, audio/video data is transmitted from the server faster than it is played out by the user system, thus building up audio/video data in the user buffer.

Having thus described the invention in rather full detail, it will be understood that such detail need not be strictly adhered to, but that additional changes and modifications may suggest themselves to one skilled in the art, all falling within the scope of the invention as defined by the subjoined claims.

I claim:

1. A media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number, comprising

a processor;

a memory;

a connection to the network; and

media player software comprising

instructions to cause the media player to request from the media source a predetermined number of media data elements;

instructions to cause the media player to receive media data elements sent to the media player by the media source and store the media data elements in the memory;

instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer;

instructions to cause the media player to play media data elements sequentially from the player buffer; and

instructions to cause the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.

2. A computer adapted to function as a media player in accordance with claim **1**.

3. A wireless phone adapted to function as a media player in accordance with claim **1**.

4. The media player of claim **1**, wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player.

* * * * *

Exhibit 10
to
Ex Parte Reexamination
of
U.S. Patent No.
8,327,011

US006005600A

United States Patent [19]

Hill

[11] **Patent Number: 6,005,600**

[45] **Date of Patent: Dec. 21, 1999**

[54] **HIGH-PERFORMANCE PLAYER FOR DISTRIBUTED, TIME-BASED MEDIA**

[75] Inventor: **Ralph D. Hill**, Los Gatos, Calif.

[73] Assignee: **Silcon Graphics, Inc.**, Mountain View, Calif.

[21] Appl. No.: **08/733,478**

[22] Filed: **Oct. 18, 1996**

[51] **Int. Cl.**[6] **H04N 7/10**; H04N 7/14

[52] **U.S. Cl.** **348/7**; 348/10; 348/13; 348/12; 455/5.1; 455/4.2

[58] **Field of Search** 348/7, 10, 13, 348/12, 6; 455/5.1, 4.2, 3.1, 6.2; 370/428, 522; H04N 7/10, 7/14

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,357,276	10/1994	Banker et al.	348/7
5,442,389	8/1995	Blahut et al.	348/7
5,477,263	12/1995	O'Callaghan et al.	348/7
5,517,257	5/1996	Dunn et al.	346/7
5,528,282	6/1996	Voeten et al.	348/7

(List continued on next page.)

OTHER PUBLICATIONS

"24 Digital TV Channels in the Space of a Single Analog Channel," Imedia: StatMux, published on the Internet at http://www.imedia.com/statmux.html, pp. 1–8, date unknown.

Cowan et al., "Adaptive Methods for Distributed Video Presentation," *ACM Computing Surveys,* vol 27, No. 4, Dec. 1995, pp. 580–583.

"Full VCR–Like Functionality in a Broadcast Environment," Imedia StatMux: Product Notes, published on the Internet at http://www.imedia.com/virtualver.html, pp. 1–2, date unknown.

"Immedia Sets MPEG–2 Record at NAB '96: Demonstrates 24 Digital TV Channels in Single Analog Channel Bandwidth," Imedia: Press Release\NAB '96, published on the Internet at http://www.imedia.com/p–nabdemo.html, pp. 1–3, date unknown.

"Increased Channel Utilization for Pre–Compressed Video, NVOD and PPV," Imedia StatMux: Produt Notes, published on the Internet at http://www.imedia.com/precompressed.html, pp. 1–2, date unknown.

"Increased Channel Utilization," Imedia StatMux: Product Notes, published on the Internet at http://www.imedia.com/realtime.html, pp. 1–2, date unknown.

McCanne et al., "vic: A Flexible Framework for Packet Video," Proceedings, ACM Multimedia '95, San Francisco, California, Nov. 5–9, 1995, pp. 511–522.

"StarCast Product Overview," published on the Internet at http://www.starlight.com/starlight/html/prodfldr/starcast.html, pp. 1–3, date unknown.

"StarWare: The High Performance Multi–Media Software," published on the Internet at http://www.starlight.com/starlight/html/prodfldr/StarWare20.html, pp. 1–4, date unknown.

"StarWorks Family Overview," published on the Internet at http://www.starlight.com/starlight/html/prodfldr/starworks3.0.html, pp. 1–3, date unknown.

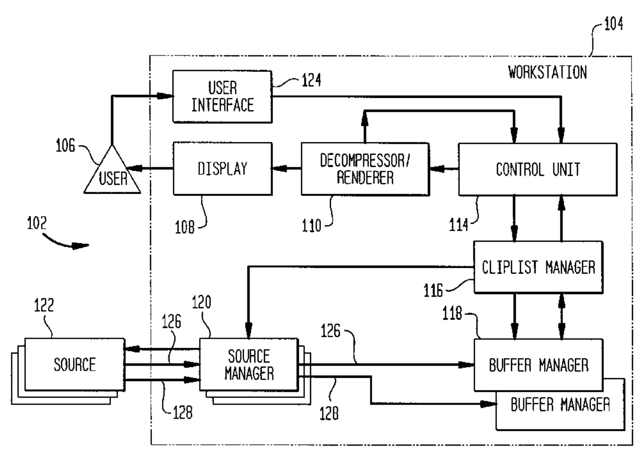
"Training on Demand," published on the Internet at http://www.starlight.com/starlight/html/prodfldr/ds.tod.frame.html, pp. 1–2, date unknown.

Primary Examiner—Nathan Flynn
Assistant Examiner—Vivek Srivastavia
Attorney, Agent, or Firm—Sterne, Kessler, Goldstein & Fox P.L.L.C.

[57] **ABSTRACT**

The present invention is directed to a system and method for enabling a user to view a moving picture with synchronized audio using a workstation connected to a digital network. The user may change both speed and direction, as well as jump between arbitrary points in the movie being played. According to the present invention, digitized movie data may be accessed from different sources (e.g., network servers, local disks). These sources vary in speed and response time as a result of their capabilities (e.g., limited I/O bandwidth) and transient events (e.g., resource contention). The present invention adapts to these varying conditions in order to maintain a stable output.

18 Claims, 7 Drawing Sheets



U.S. PATENT DOCUMENTS

5,544,327	8/1996	Dan et al.	348/7
5,606,359	2/1997	Youden et al.	348/7
5,629,732	5/1997	Moskowitz et al.	348/7
5,720,037	2/1998	Biliris et al.	348/7
5,721,815	2/1998	Ottesen et al.	348/7
5,761,303	6/1998	Hobbs et al.	380/19

FIG. 1

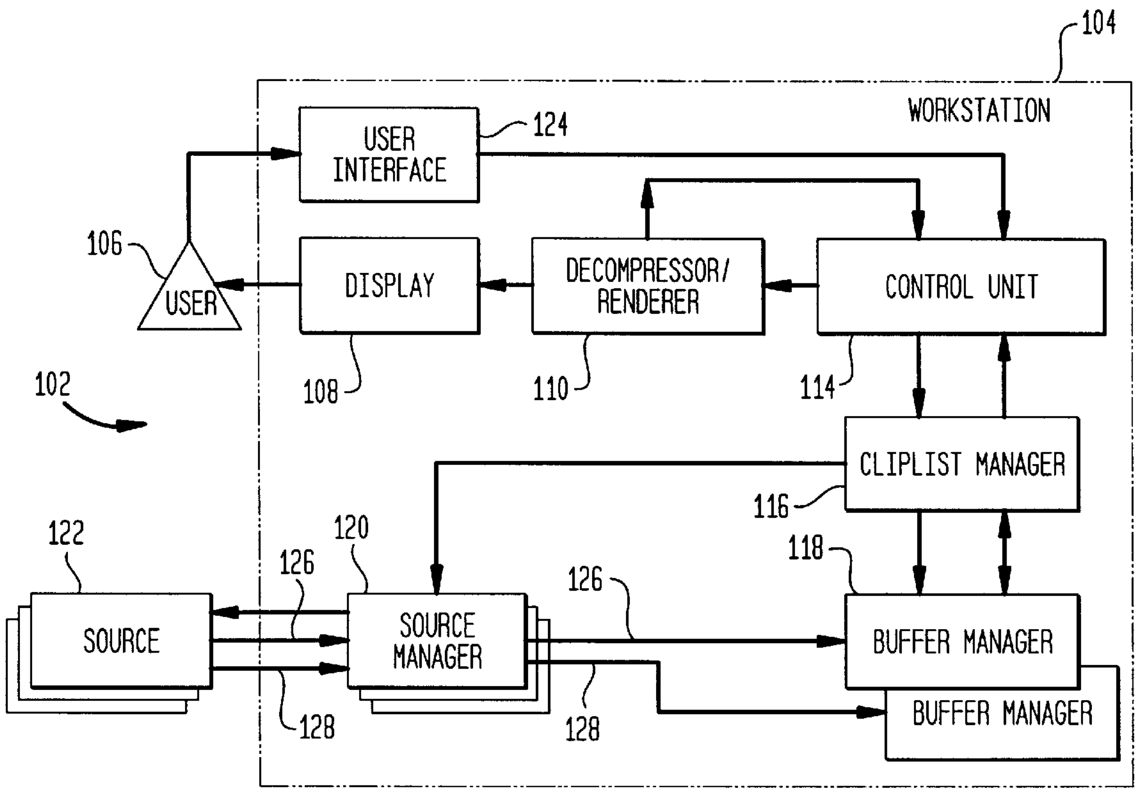
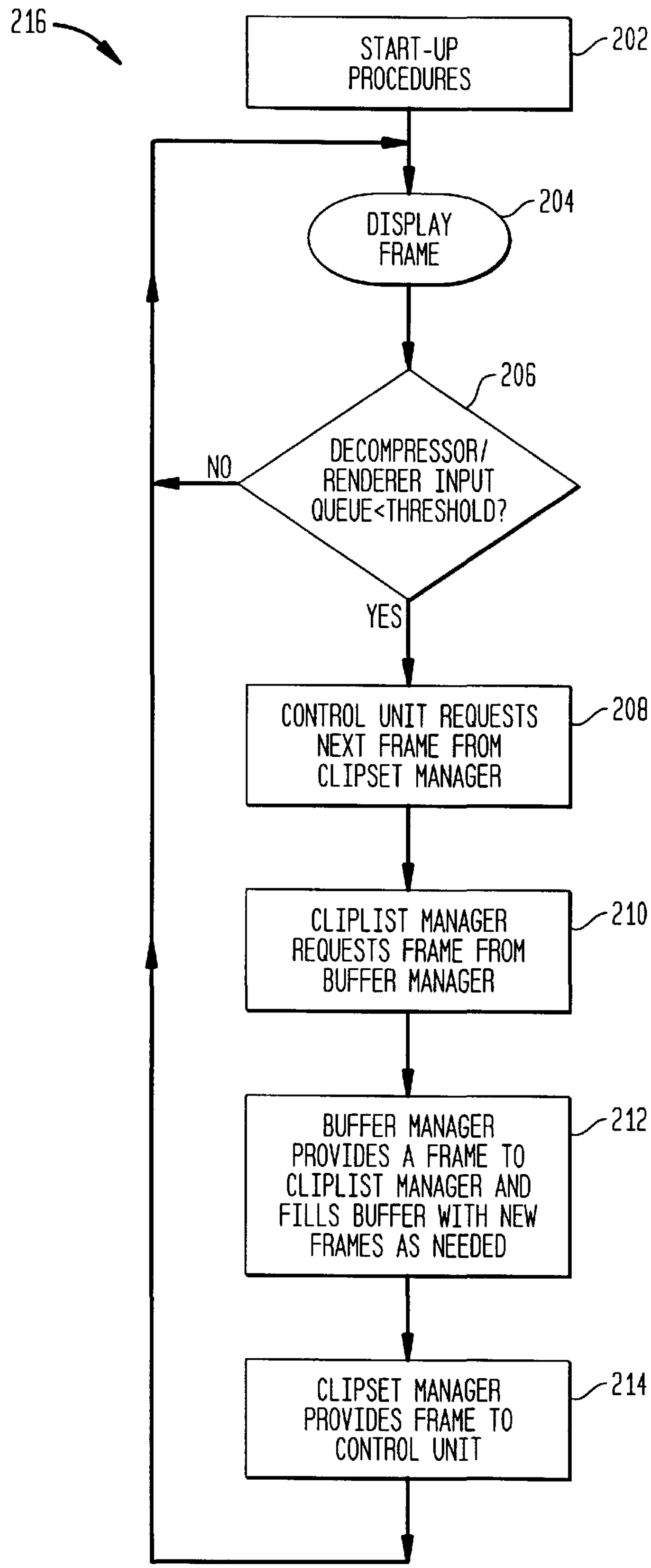

104
WORKSTATION
USER INTERFACE
124
106
USER
DISPLAY
DECOMPRESSOR/ RENDERER
CONTROL UNIT
108
110
114
102
CLIPLIST MANAGER
116
118
122
120
126
126
SOURCE
SOURCE MANAGER
BUFFER MANAGER
BUFFER MANAGER
128
128

FIG. 2



216
START-UP PROCEDURES
202
DISPLAY FRAME
204
DECOMPRESSOR/ RENDERER INPUT QUEUE<THRESHOLD?
206
NO
YES
CONTROL UNIT REQUESTS NEXT FRAME FROM CLIPSET MANAGER
208
CLIPLIST MANAGER REQUESTS FRAME FROM BUFFER MANAGER
210
BUFFER MANAGER PROVIDES A FRAME TO CLIPLIST MANAGER AND FILLS BUFFER WITH NEW FRAMES AS NEEDED
212
CLIPSET MANAGER PROVIDES FRAME TO CONTROL UNIT
214

FIG. 3

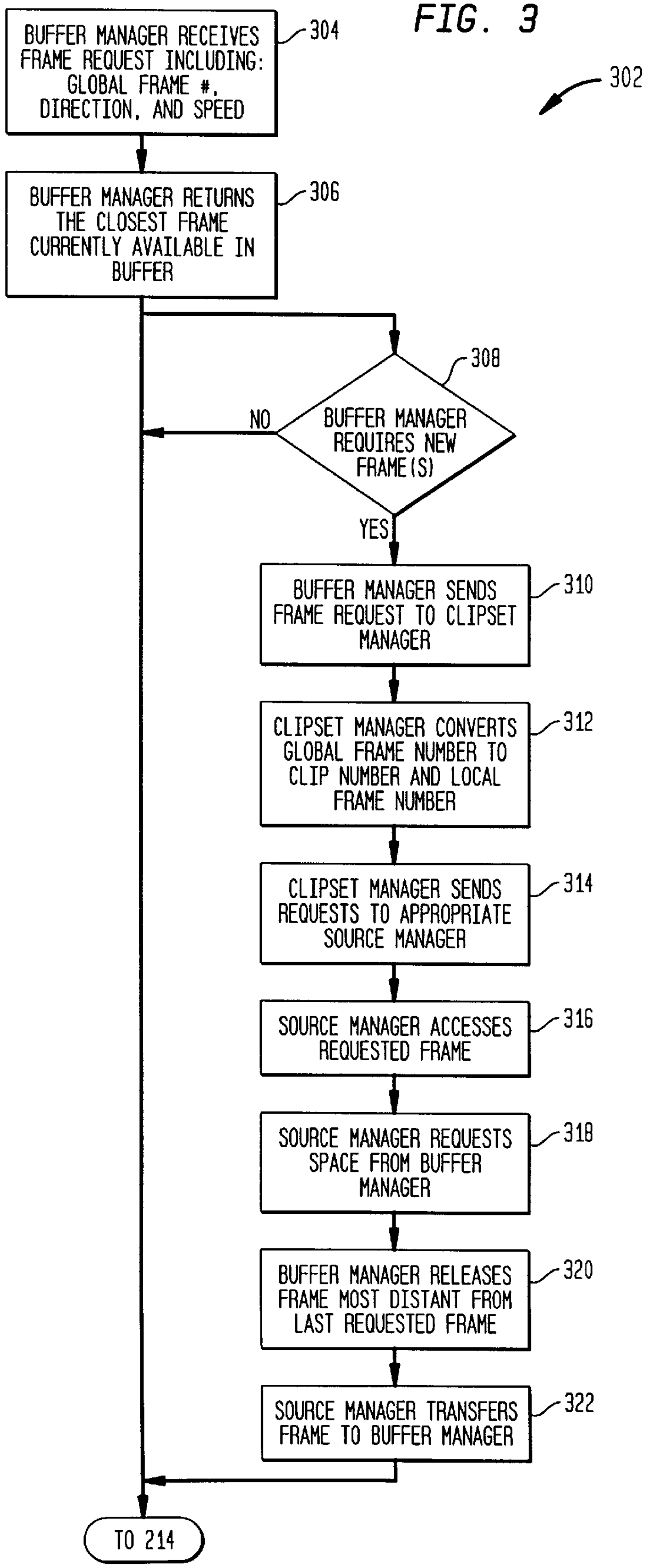

302
BUFFER MANAGER RECEIVES FRAME REQUEST INCLUDING: GLOBAL FRAME #, DIRECTION, AND SPEED
304
BUFFER MANAGER RETURNS THE CLOSEST FRAME CURRENTLY AVAILABLE IN BUFFER
306
BUFFER MANAGER REQUIRES NEW FRAME(S)
308
NO
YES
BUFFER MANAGER SENDS FRAME REQUEST TO CLIPSET MANAGER
310
CLIPSET MANAGER CONVERTS GLOBAL FRAME NUMBER TO CLIP NUMBER AND LOCAL FRAME NUMBER
312
CLIPSET MANAGER SENDS REQUESTS TO APPROPRIATE SOURCE MANAGER
314
SOURCE MANAGER ACCESSES REQUESTED FRAME
316
SOURCE MANAGER REQUESTS SPACE FROM BUFFER MANAGER
318
BUFFER MANAGER RELEASES FRAME MOST DISTANT FROM LAST REQUESTED FRAME
320
SOURCE MANAGER TRANSFERS FRAME TO BUFFER MANAGER
322
TO 214

FIG. 4A

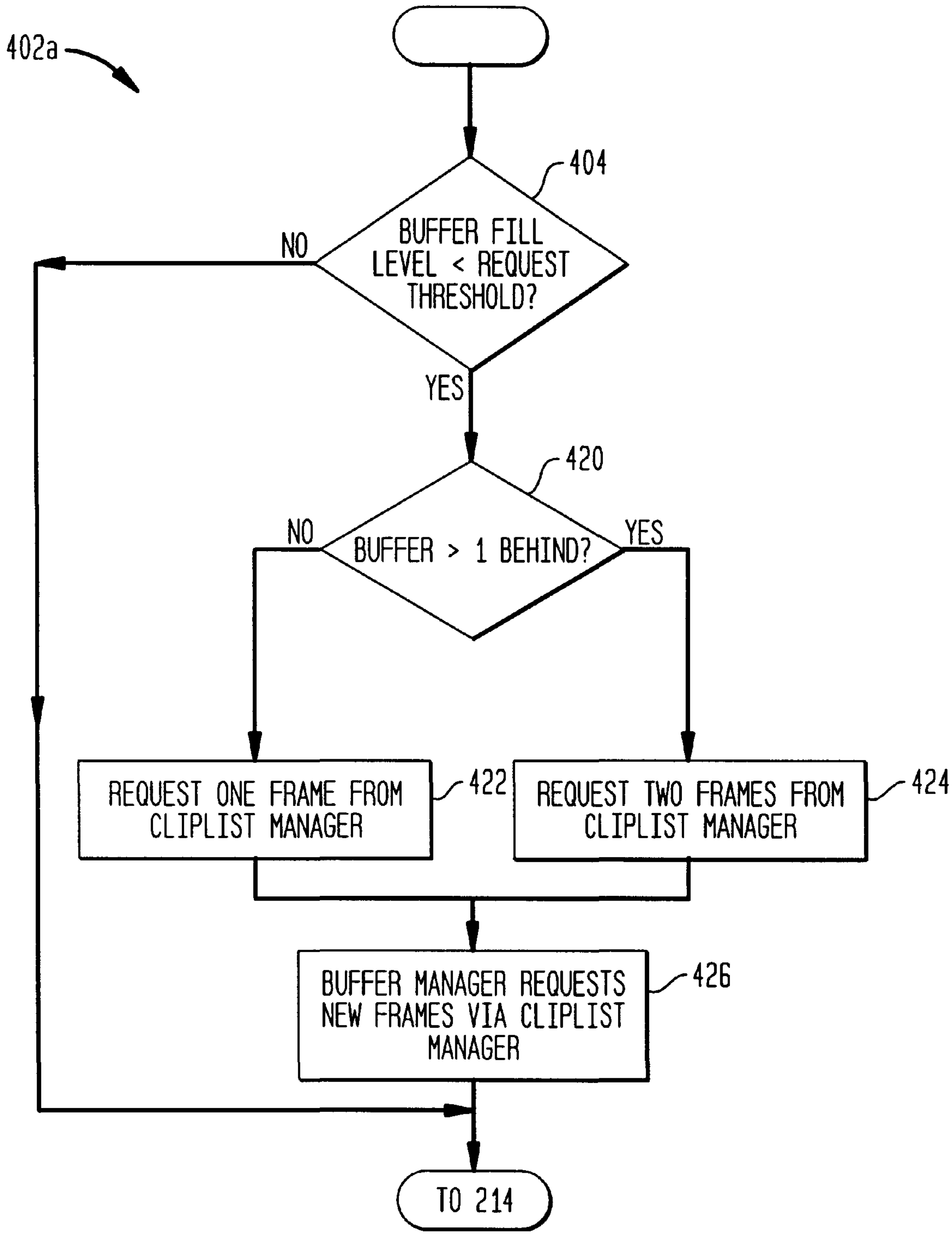

402a
404
BUFFER FILL LEVEL < REQUEST THRESHOLD?
NO
YES
420
BUFFER > 1 BEHIND?
NO
YES
REQUEST ONE FRAME FROM CLIPLIST MANAGER
422
REQUEST TWO FRAMES FROM CLIPLIST MANAGER
424
BUFFER MANAGER REQUESTS NEW FRAMES VIA CLIPLIST MANAGER
426
TO 214

FIG. 4B

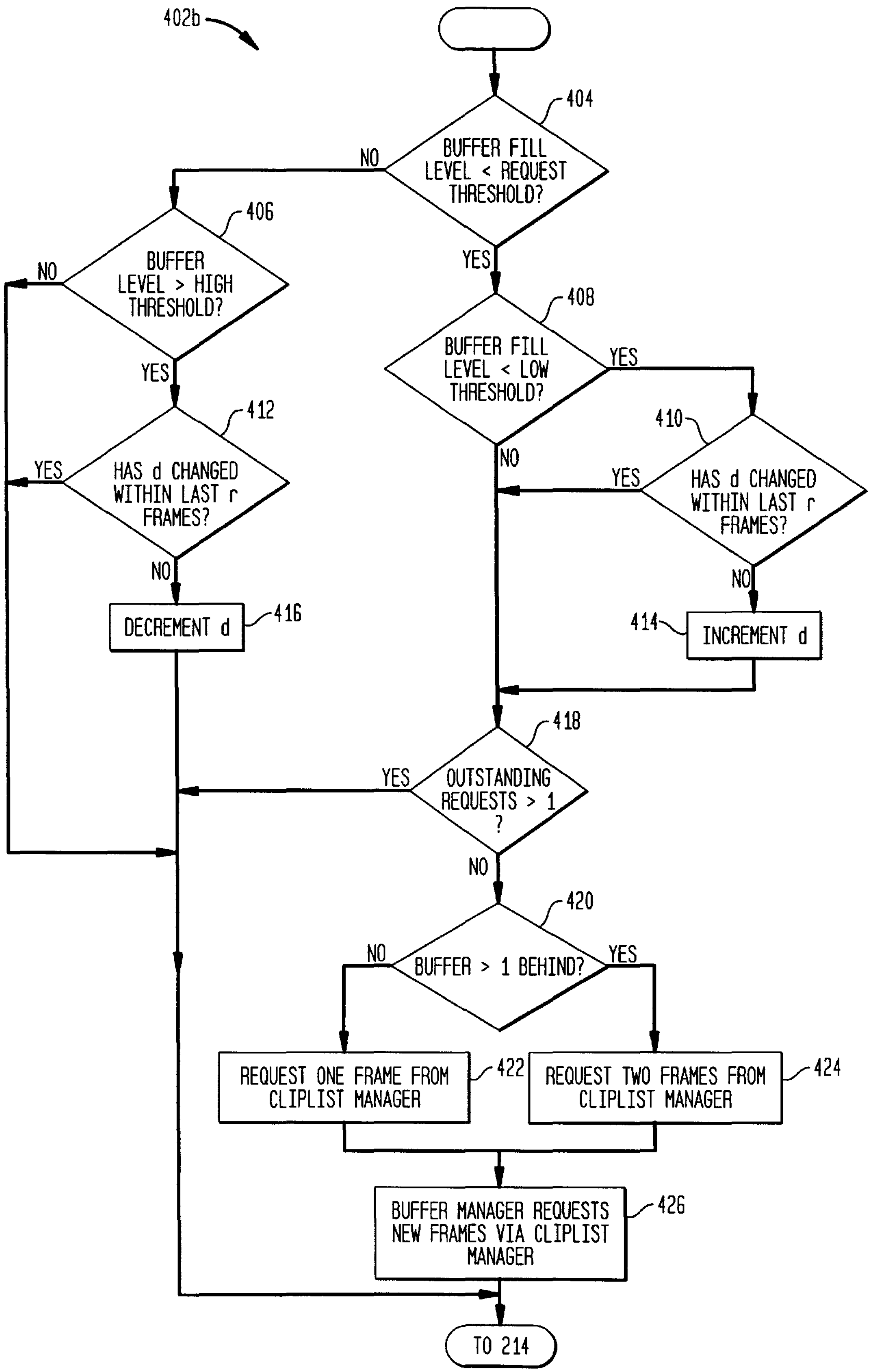

402b
404
BUFFER FILL LEVEL < REQUEST THRESHOLD?
NO
YES
406
BUFFER LEVEL > HIGH THRESHOLD?
NO
YES
408
BUFFER FILL LEVEL < LOW THRESHOLD?
YES
NO
412
HAS d CHANGED WITHIN LAST r FRAMES?
YES
NO
410
HAS d CHANGED WITHIN LAST r FRAMES?
YES
NO
416
DECREMENT d
414
INCREMENT d
418
OUTSTANDING REQUESTS > 1 ?
YES
NO
420
BUFFER > 1 BEHIND?
NO
YES
422
REQUEST ONE FRAME FROM CLIPLIST MANAGER
424
REQUEST TWO FRAMES FROM CLIPLIST MANAGER
426
BUFFER MANAGER REQUESTS NEW FRAMES VIA CLIPLIST MANAGER
TO 214

FIG. 5A

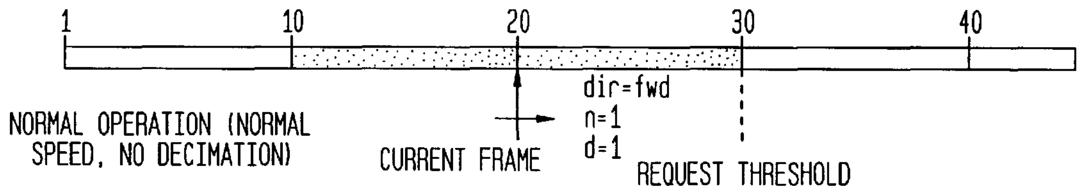


FIG. 5B

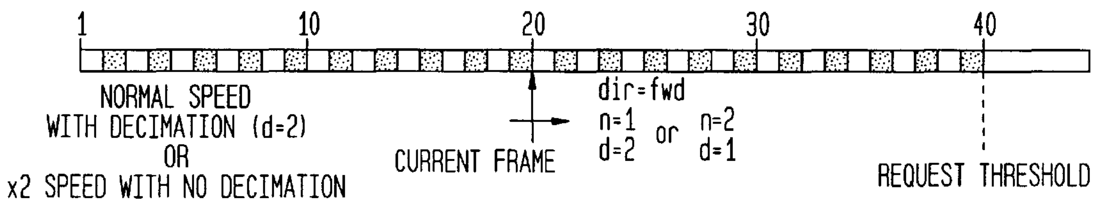


FIG. 5C

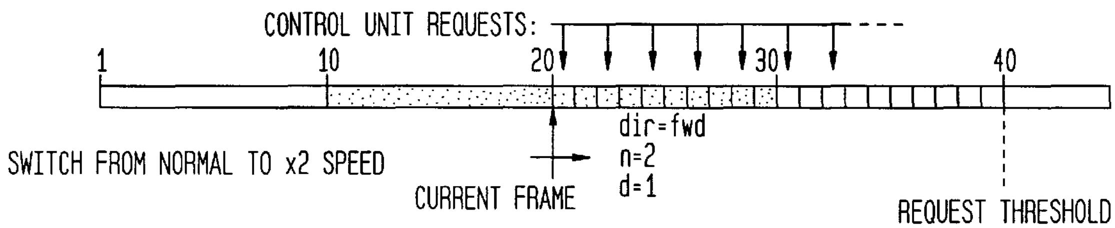


FIG. 5D

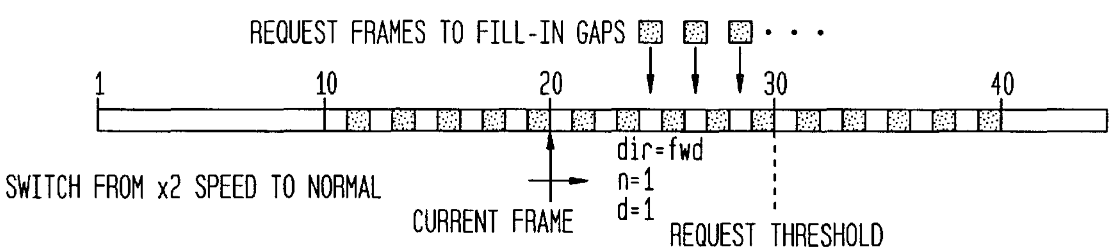


FIG. 6A

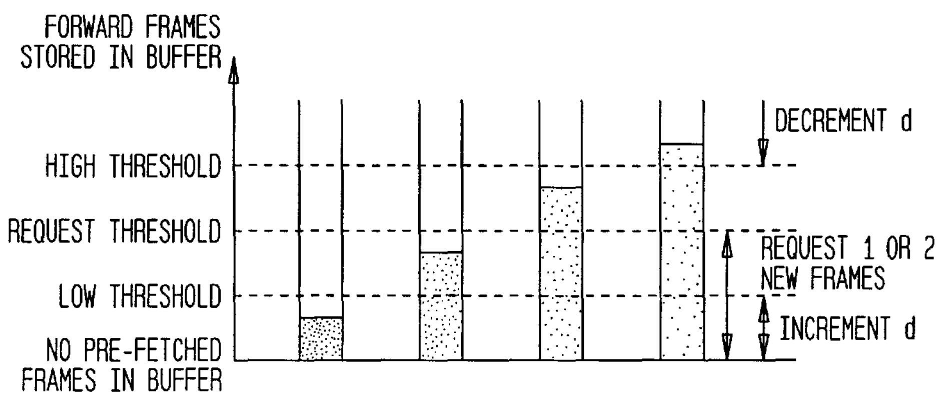


FIG. 6B

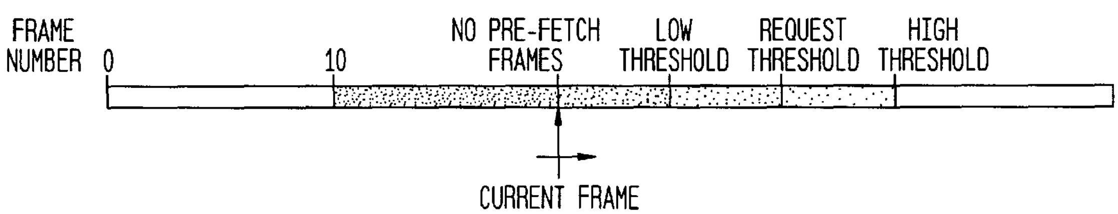
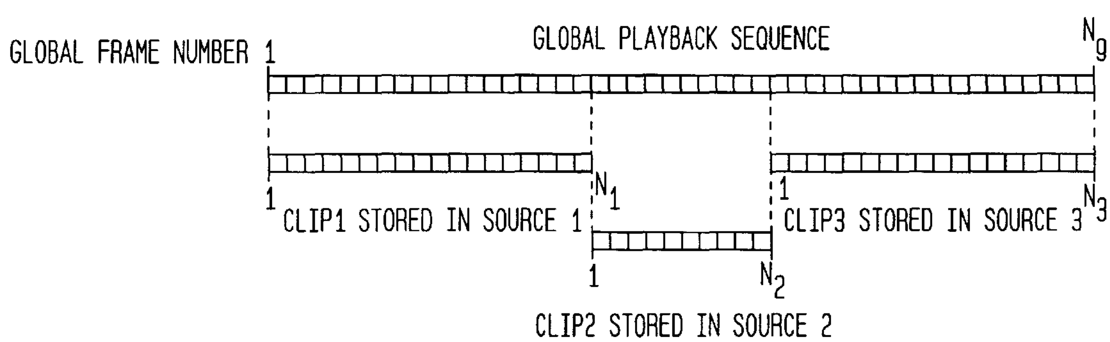


FIG. 7



HIGH-PERFORMANCE PLAYER FOR DISTRIBUTED, TIME-BASED MEDIA

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to audio/video playback devices, and more particularly to playing movie data held in various storage devices with varying bandwidth capabilities, where the movie player adapts to these varying conditions.

2. Related Art

Today, a wide variety of applications demand that a workstation display a moving image with accompanying sound. Often these workstations must access the audio and video data from multiple sources, some local and some accessed via a network connection. This data must be supplied to the display unit at a constant rate in order to achieve a steady picture. Problems arise when the data is not provided to the workstation at the required rate.

This problem is particularly acute when the data source must be accessed via a network. Delays in accessing data over standard networks tend to be longer and have a wider variance than accessing data from a local source (e.g., local RAM, local disk drive). Conventionally, one must insure that the network provides the data at a constant rate in order to achieve an acceptable output.

This solution, known as a "server push" solution, requires a powerful and complex server. The client, however, may be very simple. The client sends a request specifying what to play, and how fast to play it. The server responds by sending the requested information at the requested rate. Again, the rate at which the information flows from the server to the client must be very close to constant as the client has limited memory to buffer the information.

This approach presents several problems. Any interruption in the flow of data will cause an immediate degradation in output quality. Thus, there is no allowance for network contention or errors. Further, there is no feedback from the client to the server to advise the server of problems with the transmission. The client only sends messages to the server when the user issues a playback command, such as "stop," or "fast forward."

The primary advantage associated with the server push solution is the minimal cost of client hardware and software. However, this advantage is more than offset by the cost of a sophisticated server which must provide a guaranteed bandwidth and actively monitor the status of the client. This complexity has a deleterious effect on server reliability: increased sophistication results in decreased reliability.

A need exists for a system which offers high-performance, user-controlled playback over a general purpose computer network. Users often want to integrate this capability into an existing network which performs numerous other tasks as well. However, existing networks often are unreliable and unsophisticated, precluding the use of a server-push solution. Another solution which accommodates the use of an existing network would have obvious cost advantages over the server-push which requires a new, sophisticated network dedicated almost exclusively to audio/video playback.

Also, users have often invested heavily in relatively sophisticated client hardware, such as conventional personal computers or high-performance workstations. Again, this does not fit the server-push solution.

Numerous applications exist for the appropriate solution. Multi-media software, commonly used today in a variety of

environments, integrates moving pictures and sound. Dedicated machines are also being developed for playing games over a network. These games also involve sophisticated graphics and sound.

Finally, modern animation is often developed in a shared network environment where multiple animators access a common database. It is often the case where data needs to be accessed from different sources. Ideally, the scene currently being edited by the animator will be stored locally for easy access. In order to develop the scene in context, it is very useful to be able to play the scenes immediately before and after. These surrounding scenes, called "hook-ups", often are stored as part of a large library accessed via the network.

These hook-ups may be copied to the local machine before being played. However, this takes time and does not insure that the animator has been provided with the latest copy as editing of all scenes may be ongoing.

Thus, what is requires is an improved system and method for utilizing existing computer networks and client hardware to provide user-controlled, high performance audio/video playback.

SUMMARY OF THE INVENTION

Briefly stated, the present invention is directed to a system and method for user-controlled playback of synchronized audio and video information where the data is accessed via sources of varying bandwidth capabilities.

The system provides a buffer management system whereby the strategy for accessing data adapts as source capabilities change. The system attempts to prefetch data that is likely to be required in the near future. This buffering allows the system to provide constant picture quality in case the flow of data slows.

The present invention monitors the fill level of the buffer and if it has difficulty maintaining an acceptable level of data, the system begins to selectively make fewer requests. This allows the buffer to catch-up while easing the burden placed on the sources to provide data. As the buffer fills, the request strategy returns to normal.

The present invention also retains a certain amount of data just displayed. Having this data on hand allows the system to respond immediately to user requests for a change in direction of play.

Further features and advantages of the invention, as well as the structure and operation of various embodiments of the invention, are described in detail below with reference to the accompanying drawings. In the drawings, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements. The drawing in which an element first appears is indicated by the leftmost digit(s) in the corresponding reference number.

BRIEF DESCRIPTION OF THE FIGURES

The present invention will be described with reference to the accompanying drawings, wherein:

FIG. **1** is a block diagram of a digital network environment according to a preferred embodiment of the present invention;

FIG. **2** is a flowchart representing the general operation of the preferred embodiment;

FIG. **3** is a flowchart representing the preferred input/output operation of the buffer manager;

FIG. **4A** is a flowchart representing the basic operation of the preferred buffer management strategy;

FIG. 4B is a flowchart representing the operation of the preferred buffer management strategy including additional structures to account for abnormal operating conditions;

FIG. 5 represents the buffer management strategy given a variety of operating conditions;

FIG. 6 summarizes the various thresholds applied to the buffer fill level according to the preferred buffer management strategy; and

FIG. 7 represents the organization and representation of movie data within the preferred embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Overview of the Invention

The present invention is directed to a system and method for enabling a user to view a moving picture with synchronized audio using a workstation connected to a digital network. The user may change both speed and direction, as well as jump between arbitrary points in the movie being played. According to the present invention, digitized movie data may be accessed from different sources (e.g., network servers, local disks). These sources vary in speed and response time as a result of their capabilities (e.g., limited I/O bandwidth) and transient events (e.g., resource contention). The present invention adapts to these varying conditions in order to maintain a stable output.

FIG. **1** is a block diagram of a digital network environment **102** within which the invention is used. A user **106** utilizes the workstation **104** to play movie data. The workstation **104** may be a conventional personal computer, a high performance workstation, or dedicated hardware specially suited for audio/video playback. The user **106** views the movie data on display **108** while issuing control information (e.g., "go forward," "go backward," "play faster") via a standard user interface **124** (e.g., keyboard, mouse, joystick).

Movie data is composed of synchronized audio and video information. This information is broken into fundamental units called frames. For each video frame there is an associated audio frame. A video frame contains the video information to display a single image. An audio frame contains the audio information associated with that image. For ease of notation, the term 'frame' will be used throughout to refer to either audio or video frames, as they are generally treated the same.

A clip is a sequence of frames. The movie data of interest to a user will be composed of one or more clips.

Clips may be stored in different locations within the digital network environment **102**. For purposes of discussion, assume that the movie data of interest to the user contains N_c clips, each stored in a separate source **122**. The clips may or may not be stored according to a conventional compression format. Each source **122** may be any type of digital memory, and may be accessed either locally or via a network server. Note that a source may be a site on the Internet.

The workstation **104** contains a control unit **114** responsible for the overall control of the system. The control unit **114** receives control information from the user **106** via the user interface **124**. Examples of control information include commands to stop, reverse play, speed-up play, slow-down play, or skip to an arbitrary frame. The control unit **114** also receives control information from the decompressor/renderer **110**. This control path will be discussed below.

The display **108** provides a video image for the user **106**, while a conventional speaker (not shown) provides the audio. The decompressor/renderer **110** is responsible for decompressing the video and audio and presenting them to the display **108** in correct synchronization. The control unit **114** is responsible for feeding frames to the decompressor/renderer **110** in the proper sequence. Note that decompression is only required if the movie data is stored in a compressed format (e.g., JPEG for video information).

The buffer manager **118** provides frames upon request to the control unit **114** via the cliplist manager **116**. In order to provide frames quickly, the buffer manager **118** stores a limited number of frames in local digital memory, such as random access memory (RAM) or a local disk drive. The goal is to always have available the next frame required by the control unit **114** when that frame is requested. The buffer manager **118** accomplishes this goal by fetching frames likely to be required before they are requested. This will be referred to as prefetching future frames.

The buffer manager **118** also retains a certain number of the frames most recently sent to the control unit **114**. This allows the buffer manager **118** to respond immediately if the direction of play is reversed. These frames will be referred to as history frames.

The buffer manager **118** transfers frames to the control unit **114** via the cliplist manager **116**. It is the responsibility of the cliplist manager **116** to keep track of where within the digital network environment **102** each clip is stored. Thus, in order to replenish it's store of frames, the buffer manager **118** must request them via the cliplist manager **116**.

The cliplist manager **116** requests frames from a particular source **122** via the appropriate source manager **120**. The source manager **120** provides the necessary interface (i.e., hardware or software) for communicating with a source **122**. The digital network environment **102** includes an arbitrary number of sources (N_c) from which frames may be accessed. The workstation **104** provides one source manager **120** for each source **122**.

Movie data flows between the source **122**, the source manager **120**, and the buffer manager **118** along discrete paths, or tracks, with one track for each type of information. Two tracks will be described with respect to the preferred embodiment: audio and video. However, these are discussed by way of example only. Other types of data might also be useful. For example, in the context of professional video editing, user commentary would be a useful supplement to append to particular portions of the movie data.

Audio data flows along the audio track **126**, as shown in FIG. **1**. Similarly, video data flows along the video track **128**. Note that a separate buffer manager **118** is provided for each track. The buffer managers generally operate independently of each other, with two exceptions. First, they must provide audio and video frames to the control unit **114** synchronously so that image and audio are correctly aligned. Second, the concept of priority tracks, discussed in detail below, forces them to operate in a somewhat interdependent manner. However, unless otherwise specifically noted, the buffer managers all operate in an identical fashion, and all will be referred to as buffer manager **118**.

General Operation

FIG. **2** is a flowchart **216** representing the general operation of the present invention. The following discussion will also make references to the system depicted in FIG. **1**. In step **202**, the control unit **114** initiates start-up procedures when the user first requests playback of movie data. These procedures basically involve filing the buffer manager **118** with the appropriate frames, passing the first frame to the control unit **114,** and issuing other preparatory commands. A detailed description is provided below.

In step **204**, the control unit **114** causes the first frame to be displayed to the user by passing the data to the

decompressor/renderer **110**. The operation of the decompressor/renderer **110** is well known in the art, and as such will not be discussed in detail herein.

In step **206**, the control unit **114** monitors the input queues of the decompressor/renderer **110**. When the queue fill level drops below a certain threshold, the control unit **114** requests the next frame from the cliplist manager **116**, as shown in step **208**.

In step **210**, the cliplist manager **116** requests this frame from the buffer manager **118**. The buffer manager **118** provides a frame to the cliplist manager **116**, as shown in step **212**. The buffer manager **118** then refills its buffers with new frames as they are needed. This refill operation is critical to the present invention, and is described in detail below with respect to FIG. **3** and FIG. **4**.

In step **214**, the cliplist manager **116** gives the frame to the control unit **114**, which, in turn, gives it to the decompressor/renderer **110**. The operation continues by displaying the next frame, step **204**. Frames are displayed without interruption until the input queue in the decompressor/renderer **110** again falls below a threshold.

Buffer **1/0**

FIG. **3** is a flowchart **302** providing the detailed operation of step **212** as shown in FIG. **2**. In general, buffer manager **118** responds to requests for frames from cliplist manager **116** then determines whether new frames need to be loaded. If new frames are required, it uses the cliplist manager **116** to retrieve these frames from the appropriate source **122**. The source **122** accesses the requested frame then requests space from the buffer manager **118** before transferring it.

In step **304**, the buffer manager **118** receives a frame request from cliplist manager **116**. The request includes a global frame number (described below), the direction of play (i.e., forward or reverse), and the speed of play.

In step **306**, the buffer manager **118** returns the closest frame currently available in the buffer. Normally the exact frame is available. However, under certain circumstances this will not be the case (e.g., the buffer manager **118** was unable to retrieve a particular frame in time due to resource contention). Displaying the closest frame rather than the exact frame will cause some degradation in performance. This degradation, though, is preferable to suspending operation and displaying a blank screen.

At this point, the flowchart **302** indicates two paths. One path shows control passing back to step **214**, the other to step **308**. According to the preferred embodiment, these two paths are performed concurrently.

In step **308**, the buffer manager **118** determines whether additional future frames must be prefetched in order to have them on hand when they are later requested by the control unit **114**. This determination is described in detail below with respect to FIG. **4**. In a preferred embodiment, the buffer manager **118** either makes no request, requests one frame, or requests two frames. If the buffer manager **118** requests no frames, normal operation continues through step **214** without any further action by the buffer manager until the control unit **114** makes another request.

If the buffer manager **118** requests either one or two new frames, operation continues to step **310**. Note that frames may be referenced in one of two different ways. The control unit **114** and the buffer manager **118** reference frames according to a global frame number, as shown in FIG. **7**. Here, three clips are shown (N_c=3), each of a different length (N_1, N_2, N_3) and stored in a different source. The three clips together form the global playback sequence.

Frames may either be referenced according to their position within the global playback sequence (i.e., global frame number), or according to their position within the clip from which they originate (i.e., local frame number). The control unit **114** and the buffer manager **118** reference frames according to the former. Thus, requests from the control unit **114** may be answered by the buffer manager **118** without having to translate the frame number.

In step **310**, the buffer manager **118** sends requests for new frames to the cliplist manager **116**. The cliplist manager **116** maintains the mapping between global and local frame numbers. Thus, in step **312**, the cliplist manager **116** converts the global frame number to a clip number and local frame number.

In step **314**, the cliplist manager **116** requests the frames from the appropriate source manager **120**. The source manager **120** in step **316** retrieves the requested frame data from the source **122**.

In step **318**, the source manager **120** requests space in memory for the new frame(s) from the buffer manager **118**. The buffer manager **118** must then release a frame it is currently storing to provide space for the incoming frame, as shown in step **320**.

Many possible strategies might be used to determine which frame is released. According to a preferred embodiment, the buffer manager **118** releases the frame that is most distant from the frame that the buffer manager **118** has most recently requested. In general, the goal is to retain frames that are likely to be used soon and to release those unlikely to be needed. This simple approach works well in practice and is easily implemented.

In step **322**, the source manager **120** transfers the requested frame to the buffer manager **322**. Operation continues in step **214** with no further action by the buffer manager **118** until the control unit **114** requests another frame.

Buffer Management

FIG. **4** is a flowchart **402** representing the buffer management strategy of the present invention, and is a detailed representation of step **308** as shown in FIG. **3**. The following discussion will also involve FIG. **5** and FIG. **6**, which depict the buffer management strategy under a variety of operating conditions.

The buffer management strategy is first described under normal operating conditions, as depicted in FIG. **4**A The strategy is then described under a variety of abnormal conditions: high speed play, starting play, stopping play, reduced source bandwidth, increasing play speed, decreasing play speed, changing direction, and jumps in location. FIG. **4**B includes additional operational details relating to how these conditions are handled.

Normal Operation

First consider the case where the user is playing the movie at a normal play speed in the forward direction (i.e., normal operation). The goal of the buffer manager **118** is to retain some frames that have recently been shown to the user (i.e., history frames), and to get frames that will soon be shown to the user if the current speed and direction are maintained (i.e., future frames).

Referring to FIG. **4**A, the buffer manager **118** determines whether the buffer fill level has dropped below the request threshold, as shown in step **404**. The buffer fill level refers to the number of future frames currently being stored in the buffer manager **118**, not frames that have been requested but not yet received. The request threshold represents the optimum number of future frames stored in the buffer at any given time, and should be determined based on the characteristics of a particular system (e.g., I/O bandwidth, source latency).

The buffer manager **118** retains a number of history frames equal to the request threshold as well; the reasons for this will become apparent when reverse-play is discussed.

The buffer manager **118** will not request additional frames when the fill level exceeds the request threshold. Additional frames might be requested if the fill level drops below the threshold depending upon other factors. For the purposes of discussing normal operation, assume that these factors do not come into play and at least one additional frame will be requested when the fill level does not exceed the request threshold.

In step **420**, the buffer manager **118** determines by how many frames the request threshold exceeds the fill level. The buffer manager **118** will request one new frame from the cliplist manager **116** if the request threshold exceeds the fill level by one frame (i.e., one frame behind). If the buffer manager **118** is more than one frame behind, it will request two new frames. However, in a preferred embodiment no more than two new frames will be requested for each request received from the cliplist manager **116** in order to avoid overloading the sources. Hence, if the play rate is 24 frames per second, the sources will not be asked to supply more than 48 frames per second. However, the present invention is not limited to requesting only two frames at a time, as would be apparent to one skilled in the art.

In step **422**, the buffer manager **118** requests one new frame. Under normal operating conditions, the global frame number of the new frame is calculated by adding one to the last request made by the buffer manager **118**. In step **424**, the buffer manager requests the next two frames following the last request. The buffer manager **118** requests these frames via the cliplist manager **116** in step **426**. Operation continues with step **214**.

FIG. **5**(*a*) provides an example of the normal operation of the buffer manager **118** (i.e., forward play at normal speed). In this example, the control unit **114** has recently asked for frame 20. As was mentioned previously, the control unit **114** provides the direction and the speed of play in addition to a global frame number with every frame request.

When the control unit **114** requests global frame 20 (via the cliplist manager **116**), the buffer manager **118** replies by providing global frame 20 to control unit **114** via cliplist manager **116** and checks the fill level against the request threshold (for this example, request threshold=10). The buffer manager **118** does this by checking to see if it has frames 21 through 31. Normally, the most distant frame has not been requested. Assuming this is the case, the buffer manager **118** will then request global frame 31. This request is sent to the cliplist manager, which maps the global frame number to a clip number and local frame number and forwards the request to the appropriate source.

As mentioned above, the source manager **120** contacts the buffer manager **118** to reserve space for the new frame once it has been retrieved. In the preferred embodiment, the buffer manager **118** releases the frame that is most distant from the frame that the buffer manager has most recently requested. Here, global frame number **10** is released.

The following sections will discuss buffer management strategy under abnormal conditions (i.e., conditions other than forward play, normal speed).

High Speed Play

The sources **122** and the decompressor/renderer **110** often have difficulty displaying every frame at speeds faster than normal playback. The current invention solves this problem by showing every nth frame, where n is the absolute value of the rate of play, rounded up. So, to play at two times normal speed, every second frame is shown at the same rate (frames/second) as before. FIG. **5**(*b*) depicts this situation.

In steps **422** and **424** of FIG. **4**A, the buffer manager now adds n to the last frame requested when determining which frames to request. Note that the control unit **114** and the buffer manager **118** may get out of phase. The control unit may request frames 34, 36, 38, 40, . . . , while the buffer manager may be requesting 35, 37, 39, 41, ... In this case, the buffer manager **118** returns the closest frame it has. This will result in no loss of quality for the user, nor will it impair the user's ability to perform any task otherwise available.

Now when computing the buffer fill level, every nth frame is counted rather than every frame. The buffer manager **118** must look further into the future to meet the request threshold. As shown in FIG. **5**(*b*), the request threshold moves twice as far into the future at double speed. This alteration does not require that the buffer manager **118** operate faster or store more information; it only requires that the frames be fetched in a different order.

Note that the strategy for releasing frames from the buffer manager **118** that was discussed above continues to work with this modified strategy for requesting frames.

Inadequate Source Bandwidth

The preceding discussion assumed that the sources **122** will always be able to provide frames at the normal play rate. The buffer management strategy described thus far will operate properly with some jitter in source performance, so long as the sources **122** perform at the desired rate averaged over several frames (less than half the number of frames stored by the buffer manager **118**). However, not all sources will be able to meet this requirement. Some problems may be transitory (e.g., temporary resource contention), others will be continuous (e.g., low-speed networks).

The current invention solves this problem by modifying the buffer management strategy in several ways. These modifications are reflected in flowchart **402***b* of FIG. **4**B.

First, the buffer manager **118** monitors the number of requests for frames that are outstanding with the sources **122**. Having a large number of outstanding requests will not likely increase performance. It will, however, increase latency when the user changes direction or jumps to another clip. This is because all of the old, outstanding requests must be processed before any new requests. Thus, keeping the queue of outstanding requests short reduces latency.

Thus, the buffer manager **118** limits itself to two outstanding requests, as shown in step **418** of FIG. **4**B. This works well for configurations that have been tested. With high-latency, high-bandwidth networks, the number may have to be increased. Decreasing the number might make sense for networks with very low latency and very low bandwidth.

In step **418** of FIG. **4**B, the buffer manager **118** determines the number of outstanding requests before it requests new frames. No new requests are made if the number of outstanding requests is greater than two. Note that under these circumstances the buffer manager **118** will tend to fall behind, since it only requests new frames when the control unit **114** requests frames.

The current invention solves this problem by monitoring the buffer fill level and decimating requests from the buffer manager **118** when the fill level drops below a low threshold.

In step **408**, the buffer manager **118** determines whether the buffer fill level has dropped below the low threshold. A decimation factor d is maintained for each track. Each track begins with a decimation value of one (i.e., no decimation). The buffer manager **118** increments by one the decimation factor for a given track when the buffer fill level associated with that track drops below the low threshold, as shown in step **414**.

In the absence of decimation, the buffer manager **118** determines which frame to request by adding n (i.e., the

speed of play) to the global frame number of the last frame requested. With decimation, the buffer manager **118** determines the next request by adding n×d to the last global frame number requested, where d is the track's decimation factor. FIG. **5**(*b*) illustrates an example of this operation.

Note that this operation is functionally equivalent to high speed play (n=2) with no decimation, with one difference. Even though the buffer manager **118** begins decimating it's requests in order to catch up, the control unit **114** continues to request each frame. As described in step **306** (FIG. **3**), the buffer manager **118** returns the closest available frame. Thus, the control unit **114** displays each frame d times during periods of decimation while the buffer manager **118** catches up. By comparison, this does not occur during high speed play. Here, every nth frame is displayed once, resulting in video and sound that appears n× faster because the same number of frames are being displayed per second.

In step **406**, the buffer manager **118** determines whether the buffer fill level exceeds a high threshold. This situation might occur after a period of decimation and result in some oscillation. The buffer manager **118** decrements the decimation factor d for a given track when this occurs, as shown in step **416**. Thus, in the case of transitory problems, the decimation factor will be reduced when the problem is solved.

This basic approach to decimation does not work well. It tends to oscillate badly. To dampen the oscillations, the decimation factor is changed at most once every r requests for frames from the control unit, as illustrated by steps **410** and **412**. The value of r is a tunable parameter. Increasing r reduces the oscillations but can result in the decimation factor not increasing fast enough in response to sudden network problems. Careful tuning of r and the number of future frames stored by the buffer manager **118** can minimize the problem.

FIG. **6** graphically summarizes the various thresholds applied to the buffer fill level. FIG. **6**(*a*) shows that d is incremented by one when the fill level is less than the low threshold, and decremented by one when the fill level increases above the high threshold. Also, the buffer manager **118** requests two new frames when the fill level is two frames less than the request threshold, one frame when the fill level is one frame less than the request threshold, and no frames requested otherwise.

The buffer management strategy also treats audio and video information differently. In general, a movie consists of an audio track and a video track. The audio information is typically much smaller than the video information. Also, users are far more aware of losing audio information than video information. Thus, the buffer manager **118** decimates the video track more aggressively than the audio track.

The current invention implements this strategy by designating certain tracks as priority tracks. Audio tracks are marked as priority tracks by default. Video tracks are not normally marked as priority tracks.

When the buffer manager **118** associated with a priority track determines that it must increment or decrement the decimation factor, it increments or decrements a global decimation factor gas well. Priority tracks are never actually decimated; the decimation factor associated with a priority track is used to monitor how much that track has contributed to global decimation. Rather than decimating a priority track, all the other tracks are decimated instead in an attempt to reduce resource contention. This should allow the priority tracks to catch up.

Now, in steps **422** and **424** for non-priority tracks, the next frame requested is found by adding n×(d+g) to the number

of the last frame requested. Again, priority tracks are never decimated. The next frame is therefore found by adding n to the last frame requested, as before.

Finally, the buffer manager **118** normally uses the global frame number last requested as the starting point for determining which frame to request next. In extreme cases this frame can be further in the past than the most recent frame that the control unit **114** has requested. When this happens, the number of the frame most recently requested by the control unit **114** is used as the starting point instead.

Increasing Play Rate

FIG. **5**(*c*) illustrates the situation where the user changes speed from normal (n=1) to twice normal (n=2). The buffer manager **118** had been requesting every frame, according to normal operating conditions. With the speed increase, the control unit **114** now expects the buffer manager **118** to provide every other frame. Thus, the change in speed effectively reduces the buffer fill level by half, as the control unit **114** is no longer interested in every other frame stored in the buffer.

This may cause the buffer manager **118** to decimate until the buffer fill level stabilizes at the new play speed. This decimation may be noticeable to the user, though it may be minimized by tuning the size of the buffer and the high and low thresholds.

Decreasing Play Rate

FIG. **5**(*d*) illustrates the situation where the user changes speed from twice normal (n=2) to normal (n=1). Both the buffer manager **118** and the control unit **114** had been requesting every other frame. With the speed decrease, the control unit **114** now expects the buffer manager **118** to provide every frame.

In this situation, the buffer manager **118** must try to fill in the missing frames (i.e., infill). It can concentrate on this as there is no need to fetch frames that are further in the future; the buffer manager **118** already has closely spaced frames twice as far in the future as it needs.

The problem is to decide where to start trying to infill. If the buffer manager **118** begins by requesting the first missing frame, this may not allow the source **122** enough time to retrieve the frame before the control unit **114** requests it. Preferably, the buffer manager **118** will skip a few frames into the future to begin to infill. The exact number to skip should be tuned according to a particular configuration, and depends primarily on system latency.

The buffer manager **118** will continue to provide the closest available frame to the control manager **114** during the brief period before infilling begins. As a result, available frames will be displayed multiple times wherever new frames could not be infilled before being requested by control unit **114**. Though the user will experience some momentary degradation in picture and sound quality, this is preferable to halting operation while the buffer manager **118** infills completely.

Changing Direction

As described above, the buffer manager **118** retains the n_h frames most recently sent to the control unit **114** (i.e., history frames), where n_h equals the request threshold. Thus, the buffer manager **118** retains as many history frames as the optimum number of future frames. This situation is illustrated by FIG. **5**(*a*) and FIG. **6**(*b*).

Retaining history frames allows the current invention to change direction of play instantly without any degradation in audio or picture quality. The buffer manager **118** simply begins returning past frames when requested by the control unit **114**. The operation proceeds exactly as described above.

Note that some degradation of performance will likely result if the number of outstanding requests for new frames

is not limited. As discussed above, the buffer manager **118** preferably limits itself to one outstanding request. Allowing more causes latency problems when the direction of play is reversed. Upon reversing direction, the buffer manager **118** will immediately begin requesting new frames according to the new direction. However, these requests must wait until all the pending requests made while playing in the opposite direction are satisfied first.

This problem might be solved by providing the buffer manager **118** with the capability to abort these outstanding requests. According to the preferred embodiment, the costs in additional system complexity associated with such capability outweigh the benefits associated with reduced latency. Thus, limiting the number of outstanding requests to one is the preferable solution.

Starting

The buffer manager **118** has no frames stored when the user **106** begins using the workstation **104**. Play cannot begin until the buffer manager **118** receives the appropriate frames. At start-up, the control unit **114** advises the cliplist manager **116** of the first frame selected by the user **106**; this might be the first frame in the global playback sequence or some arbitrary point therein. The cliplist manager **116** passes this information on to the buffer manager **118**, which requests a group of frames on either side of this starting frame. This initial request may vary in size; preferably the buffer manager **118** will request as many frames as can be stored.

The delay perceived by the user **106** should be minimal. Assuming that the sources **122** are capable of delivering data at approximately the same rate that frames are display, the delay will equal the amount of play time stored in the buffer. This is an acceptable start-up delay, particularly if other tasks may be accomplished concurrently, such as other initialization procedures.

Stopping

Play may halt for a variety of reasons. The user may stop playback by issuing the appropriate command via the user interface **124**, or the control unit **114** may reach the first or last frame in the global playback sequence. As in the starting case, the cliplist manager **116** advises the buffer manager **118** of the situation. The buffer manager **118** preferably takes advantage of this opportunity to ensure that the optimum number of future and history frames are currently being stored. If frames are missing, the buffer manager **118** will immediately make the appropriate requests.

When the user **106** issues a stop command, the decompressor/renderer **110** notifies the control unit **114** of the frame currently being displayed. Due to decompression and rendering latency, this is not likely to be the frame that the control unit **114** has most recently handled. By providing this information from the decompressor/renderer **110** to the control unit **114**, the user can be given more rapid response to stop requests.

Jumps in Location

The user interface **124** allows the user **106** to issue commands such as "go to frame 42." It is unlikely that the buffer manager **118** will have frame 42 when this command is issued. Thus, jumps in location are treated like the start up case, where the first frame to be shown is the target frame of the "go to" command.

Supporting Multiple Clips

In typical usage, the current invention accesses a global playback sequence consisting of several clips. The cliplist manager **116** specifies where each clip can be found (e.g., file name on the local disk, server identification number and file name), which frame in the clip is the first to play, and

which frame in the clip is the last to play. The compilation of this data is known as the cliplist. A typical cliplist might look like:

server1/sequence1/scene2/latest.movie 50 138
local myscene3.movie 0 123
server1/sequence1/scene4/latest.movie 0 50

This example cliplist has three clips. The first clip is local frames 50 through 138 from the file "/sequence1/scene2/latest.movie". Similarly, the second clip is local frames 0 through 123 in "myscene3.movie", and the third clip is local frames 0 through 50 in "/sequence1/scene4/latest.movie". The second clip is stored locally on the user's workstation. The first and third are located on server1.

The cliplist manager **116** determines the overall length in frames of the global playback sequence (e.g., 138−50+1+124+51=264), and builds a table which maps global frame numbers to clip numbers and local frame numbers. The cliplist manager **116** also creates three source managers **120** of the appropriate types. The source managers **120** may be implemented as hardware or software, as would be well known to someone skilled in the art.

An example operation might proceed as follows. The control unit **114** requests global frame number 95 from the cliplist manager **116**, which forwards the request to the buffer manager **118**. The buffer manager **118** returns this frame (or the closest available frame) and determines whether any new frames need to be requested.

Assuming normal playback speed and a request threshold of **10**, the buffer manager **118** will check to see if global frames 96 through 106 are currently being stored, or whether they have been requested. The most likely case is that all but frame 106 will be requested or already stored. In this case, the cliplist manager **116** maps global frame 106 to clip **2**, local frame 17, then requests this frame from the second source manager **120**. Note that clip **1** occupies frames 1 through 89 of the global playback sequence, clip **2** occupies 90 through 213. Thus, global frame number 106 corresponds to local frame 17 (106−89) in clip **2**.

When that frame is received, the source manager **120** will arrange with the buffer manager **118** to have it inserted into the buffer as global frame 106.

If all of the global frames 96 through 106 have been requested or are stored in the buffer, no new frames will be requested. If more than one frame is needed, at most two will be requested.

While the preferred embodiment of the present invention has been described above, it should be understood that it has been presented by way of example only, and not limitation. Thus, the breadth and scope of the present invention should not be limited by the above-described exemplary embodiment, but should be defined only in accordance with the following claims and their equivalents.

Write Operations

The description of the present invention has thus far centered only on operations whereby movie data is read from memory and displayed. However, the foregoing discussion applies equally well to other embodiments where the movie data may be modified as it is displayed. The user interface **124** might provide editing controls for the user **106** to make alterations in the movie data. These alterations would be recorded by the control unit **114**, with the modified frame being sent back to the buffer manager **118** to replace the old frame. These, and other like embodiments would be readily apparent to someone skilled in the relevant art.

What is claimed is:

1. A system for playing digital movie data under the control of a user, wherein the movie data comprises a sequence of frames, said system comprising:

display means for displaying the movie data to said user;

control means, coupled to said display means, for providing the movie data to said display means under control of said user;

user interface means, coupled to said control means, for interacting with said user to provide said control of said user;

at least one source means that stores a portion of the movie data; and

buffer management means, coupled to said control means and to said source means, for storing a subset of the movie data and providing the movie data as needed to said control means, said buffer management means comprising:

local digital memory that stores a plurality of frames including future frames and history frames,

prefetch means for requesting future frames from said source means, wherein said prefetch means requests every dth future frame when said source means cannot provide future frames at the rate at which they are requested, and wherein d is greater than one; and

means for releasing a select frame from said local digital memory when a frame requested by said prefetch means becomes available.

2. The system of claim **1**, wherein said select frame is the frame in said subset most distant from the frame last requested by said prefetch means.

3. The system of claim **1**, wherein said user interface means comprises:

speed control means for causing said control means to provide every nth one of said frames to said display means; and

direction control means for controlling the order in which said frames are provided to said display means.

4. The system of claim **1**, further comprising:

cliplist management means, coupled to said control means and further coupled between said buffer management means and said source means, for tracking where said frames are stored within said source means and for routing requests from said prefetch means to said source means.

5. The system of claim **1**, wherein at least one of said source means comprises a digital memory accessed via a digital network.

6. The system of claim **5**, wherein said source means comprises a local disk drive.

7. The system of claim **1**, wherein said buffer management means provides a frame from said local digital memory nearest to the frame requested by said control means.

8. The system of claim **1**, wherein said prefetch means compares the buffer fill level to a request threshold and requests a select number of frames based on this comparison.

9. The system of claim **8**, wherein said prefetch means requests no frames if the buffer fill level is greater than said request threshold, one frame if the buffer fill level is equal to or one frame less than the request threshold, and two frames if the buffer fill level is two frames or more less than the request threshold.

10. The system of claim **1**, wherein said prefetch means makes fewer requests if the number of outstanding requests with said source means is greater than a threshold.

11. The system of claim **1**, wherein said buffer management means further comprises:

infill means for infilling gaps in said subset.

12. A workstation for playing digital movie data under the control of a user, wherein the movie data comprises a plurality of tracks, the plurality of tracks including at least an audio track and a video track, wherein each of the tracks comprises a sequence of synchronized frames, and wherein at least one remote source means stores a portion of the movie data, the workstation comprising:

display means for displaying the movie data to said user;

control means, coupled to said display means, for providing the movie data to said display means under control of said user;

user interface means, coupled to said control means, for interacting with said user to provide said control of said user;

a plurality of buffer management means, one for each of said tracks, coupled to said control means and to the remote source means, for storing a subset of the movie data and providing the movie data as needed to said control means, said buffer management means comprising:

local digital memory that stores a plurality of frames including future frames and history frames,

prefetch means for requesting future frames from the remote source means, and

means for releasing a select frame from said storage means when a frame requested by said prefetch means becomes available; and

cliplist management means, coupled to said control means and further coupled between said buffer management means and the remote source means, for tracking where said frames are stored within the remote source means and for routing requests for future frames from said prefetch means to the remote source means.

13. The workstation of claim **12**, wherein said prefetch means comprises:

retrieval means for selectively requesting future frames;

decimation means for causing said retrieval means to request every dth future frame when the remote source means cannot provide movie data at the rate requested; and

outstanding request means for causing said retrieval means to make fewer requests on the basis of the number of outstanding requests with the remote source means.

14. The workstation of claim **13**, wherein said decimation means comprises:

designation means for designating certain of said tracks as priority tracks;

decimation counter means for incrementing a decimation counter d when the buffer fill-level drops below a low threshold and decrementing said decimation counter d when the buffer fill-level exceeds a high threshold;

global decimation counter means for incrementing a global decimation counter g each time a decimation counter d is incremented for one of said priority tracks, and decrementing said global decimation counter g each time a decimation counter d is decremented for one of said priority tracks; and

means for causing said retrieval means to request every future frame for priority tracks, and every (d+g)th future frame for tracks not designated as priority tracks.

15. A method for playing digital movie data under the control of a user, wherein the movie data comprises a sequence of frames, and wherein at least one source stores a portion of the movie data, comprising the steps of:

(a) storing a plurality of frames including future frames and history frames in local storage, wherein said future

frames and said history frames are a subset of the movie data, and wherein a buffer fill level indicates the degree to which said future frames and said history frames fill the local storage;

(b) tracking the location of the frames comprising the movie data within the one or more sources;

(c) requesting future frames from the one or more sources, wherein said step of requesting includes the steps of:

(I) requesting a first number of future frames based on a comparison of said buffer fill level to a request threshold, and

(ii) requesting every dth future frame when said source means cannot provide future frames at the rate at which they are requested, and wherein d is greater than one; and

(d) displaying said locally stored frames under the control of the user.

16. The method of claim **15**, wherein said first number is zero if said buffer fill level is greater than said request threshold, wherein said first number is one if said buffer fill level is equal to or one frame less than said request threshold, and wherein said first number is two if said buffer fill level is two frames or more less than said request threshold.

17. A system for playing digital movie data under the control of a user, wherein the movie data comprises a sequence of frames, said system comprising:

display means for displaying the movie data to said user;

control means, coupled to said display means, for providing the movie data to said display means under control of said user;

user interface means, coupled to said control means, for interacting with said user to provide control to said user;

at least one source means that stores a portion of the movie data; and

buffer management means, coupled to said control means and to said source means, for storing a subset of the movie data and providing the movie data as needed to said control means, said buffer management means comprising:

local digital memory that stores a plurality of frames including future frames and history frames, wherein a buffer fill level indicates the degree to which said future frames and said history frames fill said local digital memory;

prefetch means for requesting every dth future frame from said source means, wherein said prefetch means increments d when said buffer fill-level drops below a low threshold and decrements d when said buffer fill-level exceeds a high threshold; and

means for releasing a select frame from said local digital memory when a frame requested by said prefetch means becomes available.

18. A method for playing digital movie data under the control of a user, wherein the movie data comprises a sequence of frames, and wherein at least one source stores a portion of the movie data, comprising the steps of:

(a) storing a plurality of frames including future frames and history frames in local storage, wherein said future frames and said history frames are a subset of the movie data, and wherein a buffer fill level indicates the degree to which said future frames and said history frames fill the local storage;

(b) tracking the location of the frames comprising the movie data within the one or more sources;

(c) requesting future frames from the one or more sources, wherein said step of requesting includes the steps of:

(I) requesting a first number of future frames based on a comparison of said buffer fill level to a request threshold, and

(ii) requesting every dth future frame, wherein d is incremented when said buffer fill-level drops below a low threshold, and wherein d is decremented when said buffer fill-level exceeds a high threshold; and

(d) displaying said locally stored frames under the control of the user.

* * * * *

Exhibit 12
to
Ex Parte Reexamination
of
U.S. Patent No.
8,327,011

Trials@uspto.gov
571-272-7822

Paper 28
Entered: July 16, 2020

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

WEBPOWER, INC.,

FRIENDFINDER NETWORKS INC., STREAMRAY INC., WMM, LLC, WWM HOLDINGS, LLC, and MULTIMEDIA, LLC,

DUODECAD IT SERVICES LUXEMBOURG S.À R.L., ACCRETIVE TECHNOLOGY GROUP INC., ICF TECHNOLOGY, INC., RISER APPS LLC, and STREAMME, INC. (f/k/a VUBEOLOGY, INC.),

Petitioner,

v.

WAG ACQUISITION, LLC,
Patent Owner.

Case IPR2016-01238
Patent 8,122,141 B2

Before TREVOR M. JEFFERSON, BRIAN J. McNAMARA, and PATRICK M. BOUCHER, *Administrative Patent Judges*.

BOUCHER, *Administrative Patent Judge*.

JUDGMENT
Final Written Decision on Remand
Determining All Challenged Claims Unpatentable
35 U.S.C. §§ 144, 318(a) and 37 C.F.R. § 42.73

I. INTRODUCTION

We address this case on remand after a decision by the U.S. Court of Appeals for the Federal Circuit in *WAG Acquisition, LLC v. Webpower, Inc.*, 781 F. App'x 1007 (Fed. Cir. 2019).

A. Background

In response to a Petition (Paper 1, "Pet.") filed by WebPower, Inc., we instituted an *inter partes* review of claims 10–23 of U.S. Patent No. 8,122,141 B2 ("the '141 patent"). Paper 7, 22–23. We subsequently joined FriendFinder Networks Inc., Steamray Inc., WWM, LLC, WWM Holdings, LLC, Multi Media, LLC, Duodecad IT Services Luxembourg S.à r.l., Accretive Technology Group, Inc., ICF Technology, Inc., Riser Apps LLC, and StreamMe, Inc. (f/k/a Vubeology, Inc.) as parties to the proceeding. Papers 12, 13. We refer collectively to all petitioners herein as "Petitioner."

During the trial, WAG Acquisition, LLC ("Patent Owner") timely filed a Response (Paper 11, "PO Resp."), to which Petitioner timely filed a Reply (Paper 15, "Reply"). An oral hearing was held on September 25, 2017, and a copy of the transcript was entered into the record. Paper 21 ("Tr.").

Following consideration of the fully developed record, we issued a Final Written Decision in which we concluded that Petitioner had shown, by a preponderance of the evidence, that claims 10–23 of the '141 patent are

unpatentable. Paper 22 ("Dec."). Patent Owner appealed our Decision to the Federal Circuit "as to claims 10–18." *WAG Acquisition*, 781 F. App'x at 1008. The Federal Circuit vacated our Decision "as to the appealed claims" and remanded for further consideration whether claims 10–18 are unpatentable in light of the Court's construction of a disputed limitation recited in independent claim 10. *Id.* Because Patent Owner did not appeal the Board's prior conclusion that claims 19–23 are unpatentable, those claims and related issues are not before the Board on remand. *See id.* at 1009 n.2.

On remand, the parties jointly proposed submission of simultaneous briefs addressing the patentability of claims 10–18 in light of the Federal Circuit's decision, and we adopted this procedure. Paper 25. Accordingly, Petitioned filed a Petitioner's Brief on Remand (Paper 26, "Pet. Remand Br.") and Patent Owner filed a Patent Owner's Supplemental Brief After Remand (Paper 27, "PO Remand Br.").

For the reasons discussed below, we conclude, in view of the Federal Circuit's claim construction, and a full record that includes the parties' remand briefs, that Petitioner shows, by a preponderance of the evidence, that claims 10–18 are unpatentable.

B. The '141 Patent

The '141 patent describes a system for streaming media, such as audio or video, via the Internet with reduced playback interruptions. Ex. 1001, 4:39–44. A number of factors can affect the continuity of streaming media, including the quality of a user's connection with the Internet, variations in Internet traffic that may cause congestion at various points along the route

that data flows, and the dropping of data packets by overloaded routers. *Id.* at 2:10–30. The '141 patent describes a buffering system for streaming media that seeks to limit such deficiencies. *Id.* at 4:33–35.

Figure 1 of the '141 patent is reproduced below.

Fig. 1

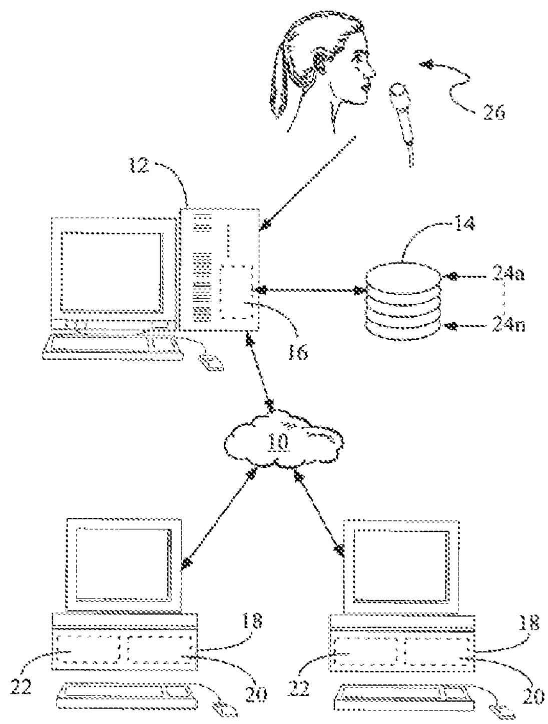


Figure 1 is a schematic diagram that illustrates elements of a streaming media buffering system. *Id.* at 10:7–9. Server 12 is connected to the Internet for transmitting sequenced streaming-media data elements. *Id.* at 10:22–25. Associated with server 12 are buffer manager 16 and first-in–first-out ("FIFO") buffer 14, which stores at least one of the data elements for transmission. *Id.* at 10:25–27. Buffer manager 16 receives the media data, supplies the media data in order to FIFO buffer 14, and maintains pointers 24a–24n into the buffer for user computers, indicating the last media data element that has been sent to respective users and thus indicating the next element or elements to be sent. *Id.* at 10:30–38. Once FIFO buffer

14 is full, the oldest data elements in the buffer are deleted as new elements are received. *Id.* at 10:38–40. A predetermined number of data elements are kept in FIFO buffer 14. *Id.* at 10:40–41.

At least one user computer 18 is connected to server 12 via the Internet. *Id.* at 10:45–46. User buffer 20 is associated with user computer 18 and stores a predetermined number of the media data elements. *Id.* at 10:47–49. Buffer manager 22, associated with user computer 18, receives and stores a predetermined number of media data elements received by the media player, plays the data out sequentially as audio and/or video, and deletes media data elements from buffer 20 as they are played out to approximately maintain the predetermined number of data elements in the user's buffer. *Id.* at 10:53–59, 8:31–34.

In an alternative embodiment, buffer manager 22 (or the media source) provides for sequentially numbering the media data elements and does not maintain a pointer into buffer 20 for each user. *Id.* at 8:38–40. "Instead, the media player buffer manager in the user computer maintains a record of the serial number of the last data element that has been received." *Id.* at 8:40–42. By using standard data communications protocol techniques, "such as TCP," user computer 18 transmits requests to server 12 for data elements specified by their serial numbers. *Id.* at 8:42–46. Server 12 responds with the requested data elements, depending "upon the reliable transmission protocol" to assure delivery, with user computer 18 then continuing with additional data requests for the duration of playing the streamed material. *Id.* at 8:46–50. "In this manner, the user computer, not the server, maintains the record of the highest data element number stored in the user computer buffer." *Id.* at 8:50–52.

C. Illustrative Claim

Independent claim 10 is illustrative of the claims at issue, and is reproduced below.

> 10. A server for distributing streaming media via a data communications medium such as the Internet to at least one user system of at least one user, the streaming media comprising a plurality of sequential media data elements for a digitally encoded audio or video program, said user system being assumed to have a media player for receiving and playing the streaming media on said user system, which is operable to obtain media data elements from said server by transmitting requests to said server to send one or more specified media data elements, said server comprising
>
> at least one data storage device, memory for storing machine-readable executable routines and for providing a working memory area for routines executing on the server, a central processing unit for executing the machine-readable executable routines, an operating system, at least one connection to the communications medium, and a communications system providing a set of communications protocols for communicating through said at least one connection;
>
> a machine-readable, executable routine containing instructions to cause the server to assign serial identifiers to the sequential media data elements comprising the program;
>
> a machine-readable, executable routine containing instructions to cause the server to receive requests from the user system for one or more media data elements specifying the identifiers of the requested data elements; and
>
> a machine-readable, executable routine containing instructions to cause the server to send media data elements to the user system responsive to said requests, at a rate more rapid than the rate at which said streaming media is played back by a user.

Id. at 13:63–14:28.

D. Grounds of Unpatentability

In relevant part, Petitioner relies on the following references. Pet. 8–10.

Carmel US 6,389,473 B1 May 14, 2002 Ex. 1003

International Standard ISO/IEC 11172-1, *Information Technology—Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s—Part 1: Systems* (ISO/IEC, August 1993) (Ex. 1018) ("ISO-11172-1")

International Standard ISO/IEC 11172-2, *Information Technology—Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s—Part 2: Video* (ISO/IEC, August 1993) (Ex. 1019) ("ISO-11172-2")

International Standard ISO/IEC 11172-3, *Information Technology—Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s—Part 3: Audio* (ISO/IEC, August 1993) (Ex. 1020) ("ISO-11172-3")[1]

The following challenges are at issue. *See* Pet. 5.

Claim(s) Challenged	35 U.S.C. §	References
10, 11, 13–18	102(a), 102(e)	Carmel
12	103(a)	Carmel and ISO-11172

[1] In its challenges, Petitioner refers collectively to ISO-11172-1, ISO-11172-2, and ISO-11172-3 as "ISO-11172." Because the challenges involving these references are all under 35 U.S.C. § 103(a), and because their description of the same standard provides a self-evident reason to combine their teachings, we do not address whether they are properly considered as a single reference or as three separate references.

E. Real Parties in Interest and Related Proceedings

In addition to the parties identified in the caption, real parties in interest with one or more of the petitioners include Various, Inc., Interactive Network, Inc., DataTech Global, LLC, DataTech Systems, LLC, Docler Media, LLC, Docler Holding S.à r.l., Gattyàn Family Irrevocable Trust (including Mr. György Gattyàn in his capacity as Grantor and Investment Advisor), Duodecad IT Services Hungary KFT, WebMindLicenses KFT, and Gattyàn Group S.à r.l. Pet. 2; *FriendFinder Networks Inc. et al. v. WAG Acquisition, LLC*, Case No. IPR2017-00786, Paper 2, 1–2; *Duodecad IT Services Luxembourg S.à r.l. v. WAG Acquisition, LLC*, Case No. IPR2017-00820, Paper 2, 2. Patent Owner identifies only itself as a real party in interest. Paper 4, 2.

The parties identify the following matters as involving the '141 patent: (1) *WAG Acquisition, LLC v. Sobonito Investments, Ltd.*, No. 2A14-cv-1661-ES-MAH (D.N.J.); (2) *WAG Acquisition, LLC v. Multi Media, LLC*, No. 2:14-cv-2340-ES-MAH (D.N.J.); (3) *WAG Acquisition, LLC v. Data Conversions, Inc.*, No. 2:14-cv-2345-ES-MAH (D.N.J.); (4) *WAG Acquisition, LLC v. Flying Crocodile, Inc.*, No. 2:14-cv-2674-ES-MAH (D.N.J.); (5) *WAG Acquisition, LLC v. Gattyàn Group S.à r.l.*, No. 2:14-cv-2832-ES-MAH (D.N.J.); (6) *WAG Acquisition, LLC v. FriendFinder Networks Inc.*, No. 2:14-cv-3456-ES-MAH (D.N.J); (7) *WAG Acquisition, LLC v. Vubeology, Inc.*, No. 2:14-cv-4531-ES-MAH (D.N.J.); (8) *WAG Acquisition, LLC v. Gamelink Int'l Ltd.* No. 2:15-cv-3416-ES-MAH (D.N.J.); (9) *WAG Acquisition LLC v. WebPower, Inc.*, No. 2:15-cv-3581-ES-MAH (D.N.J.); and (10) *WAG Acquisition, LLC v. MFCXY, Inc.*, No. 2:14-cv-3196-ES-MAH (D.N.J.). Pet. 2, Paper 4, 2–3.

The '141 patent is also the subject of IPR2015-01037, and a continuation of the '141 patent, U.S. Patent No. 8,327,011 B2, is the subject of IPR2015-01033 and IPR2016-01161. The petitions for institution of an *inter partes* review were denied for each of those proceedings. In addition, two other related patents were the subject of further *inter partes* review proceedings: (1) U.S. Patent No. 8,185,611 B2 was the subject of IPR2015-01035 and IPR2016-01162, both of whose petitions for institution of an *inter partes* review were denied; and (2) U.S. Patent No. 8,364,836 was the subject of IPR2015-01036, for which a final written decision was issued by the Board on October 20, 2016.

II. ANALYSIS

A. Claim Construction

In an *inter partes* review proceeding based on a petition filed prior to November 13, 2018, the Board interprets claims of an unexpired patent using the broadest reasonable construction in light of the specification in which they appear. *See* 37 C.F.R. § 42.100(b) (2017)[2]; *Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131, 2144–46 (2016) (upholding the use of the broadest reasonable interpretation standard).

Petitioner asserts that, in this proceeding, "no constructions are necessary," and "proposes . . . that all claim terms of the '141 patent take on their ordinary and customary meaning that the terms would have to one of

---

[2] A recent amendment to this regulation does not apply here because the Petition was filed before November 13, 2018. *See* Changes to the Claim Construction Standard for Interpreting Claims in Trial Proceedings Before the Patent Trial and Appeal Board, 83 Fed. Reg. 51,340 (Oct. 11, 2018) (amending 37 C.F.R. § 42.100(b) effective November 13, 2018).

ordinary skill in the art." Pet. 10. Patent Owner does not expressly address claim construction in its Response. Nevertheless, we determined in our Final Written Decision that it was necessary to construe the term "rate," which is recited in independent claim 10 as part of the limitation "a machine-readable, executable routine containing instructions to cause the server to send media data elements to the user system responsive to said requests, at a *rate* more rapid than the *rate* at which said streaming media is played back by the user" (emphases added). Dec. 9–10.

The context provided by our prior construction, and the Federal Circuit's disagreement with that construction, is relevant to the parties' arguments on remand. In particular, we construed rate as an "overall rate" such as might be achieved with multiple links over which data elements are sent to the recited user system. *Id.* The Federal Circuit instead construed rate as "the rate at which each requested data element is transmitted from the server to the user computer." *WAG Acquisition*, 781 F. App'x at 1011. In so construing the term, the Federal Circuit distinguished its construction as excluding the "overall rate" of our earlier construction: "The rate limitation in claim 10 therefore refers to the rate at which *requested* media data elements are sent, not the overall rate at which data is transmitted from the server to the user computer." *Id.* at 1012.

We adopt the Federal Circuit's construction for this Decision.

B. Legal Principles

Petitioner makes both anticipation and obviousness challenges. A claim is unpatentable as anticipated under 35 U.S.C. § 102 if a single prior-art reference expressly or inherently describes each limitation set forth in the

claim. *See Perricone v. Medicis Pharm. Corp.*, 432 F.3d 1368, 1375 (Fed. Cir. 2005); *Verdegaal Bros., Inc. v. Union Oil Co. of Cal.*, 814 F.2d 628, 631 (Fed. Cir. 1987).

A claim is unpatentable for obviousness under 35 U.S.C. § 103 if the differences between the claimed subject matter and the prior art are "such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations, including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) objective evidence of non-obviousness, i.e., secondary considerations.[3] *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966).

Additionally, the obviousness inquiry typically requires an analysis of "whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue." *KSR*, 550 U.S. at 418 (citing *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006) (requiring "articulated reasoning with some rational underpinning to support the legal conclusion of obviousness")); *see In re Warsaw Orthopedic, Inc.*, 832 F.3d 1327, 1333 (Fed. Cir. 2016) (citing *DyStar Textilfarben GmbH & Co. Deutschland KG v. C. H. Patrick Co.*, 464 F.3d 1356, 1360 (Fed. Cir. 2006)).

To prevail on its challenges, Petitioner must demonstrate by a preponderance of the evidence that the claims are unpatentable. 35 U.S.C.

[3] The parties do not address secondary considerations, which, accordingly, do not form part of our analysis.

§ 316(e); 37 C.F.R. § 42.1(d). "In an [*inter partes* review], the petitioner has the burden from the onset to show with particularity why the patent it challenges is unpatentable." *Harmonic Inc. v. Avid Tech., Inc.* 815 F.3d 1356, 1363 (Fed. Cir. 2016) (citing 35 U.S.C. § 312(a)(3) (requiring *inter partes* review petitions to identify "with particularity . . . the evidence that supports the grounds for the challenge to each claim")). This burden never shifts to Patent Owner. *See Dynamic Drinkware, LLC. v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015) (citing *Tech. Licensing Corp. v. Videotek, Inc.*, 545 F.3d 1316, 1326–27 (Fed. Cir. 2008)) (discussing the burden of proof in *inter partes* review). Furthermore, Petitioner does not satisfy its burden of proving obviousness by employing "mere conclusory statements." *In re Magnum Oil Tools Int'l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016).

C. Level of Skill in the Art

Petitioner's declarant, Nathaniel Polish, Ph.D., asserts that a person of ordinary skill in the art "would have had a B.S. degree in computer science or electrical engineering (or comparable degree) and two years of experience in networking or streaming media, or a M.S. in computer science or electrical engineering (or comparable degree)." Ex. 1005 ¶ 21. Dr. Polish further states that "[t]hese descriptions are approximate, and a higher level of education or specific skill might make up for less experience, and vice-versa." *Id.* ¶ 22.

Neither Patent Owner nor its declarant, Mung Chiang, Ph.D., proffers a characterization of the education and experience of a person of ordinary skill, although Dr. Chiang attests that his own qualifications permit him to

provide an opinion, "including what a person having ordinary skill in the art would have understood." Ex. 2001 ¶ 10.

We find Dr. Polish's statement of the level of ordinary skill in the art reasonable, and adopt it for this Final Written Decision.

D. Carmel

Carmel describes a method for streaming live or prerecorded media from a server to multiple client computers over the Internet. Ex. 1003, 2:1–21. Figure 2 of Carmel is reproduced below.

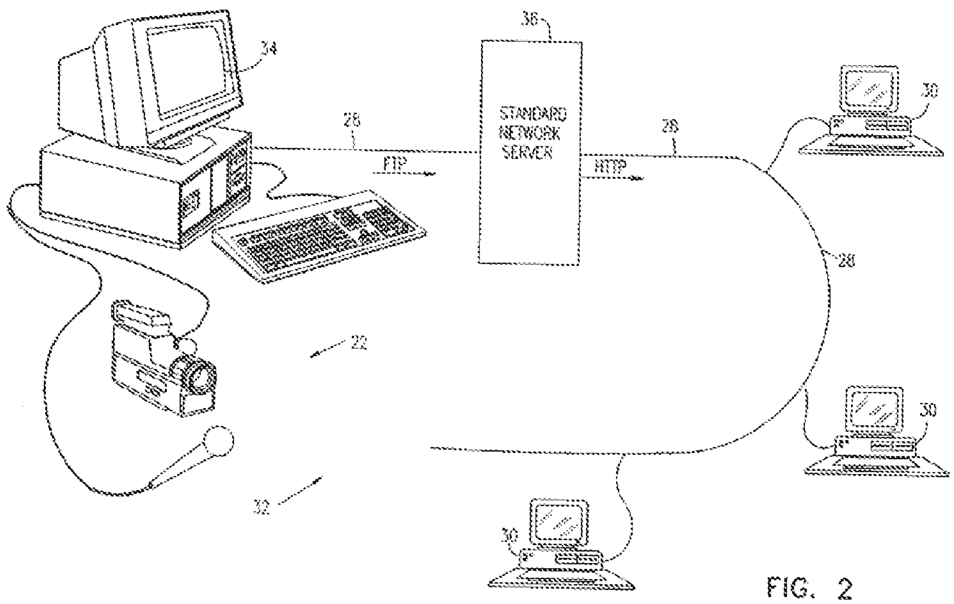


FIG. 2

Figure 2 is a schematic illustration of a computer broadcast network. *Id.* at 5:43–45. System 32 comprises transmitting computer 34 (which receives audiovisual input from devices 22), a plurality of clients 30, and network server 36, all of which communicate over network 28. *Id.* at 6:28–35. After preparing a multimedia sequence, computer 34 uploads the sequence over network 28, thereby allowing clients 30 connected with server 36 to receive the multimedia sequence in substantially real time. *Id.* at 6:50–7:17.

Figure 3A of Carmel is reproduced below.

FIG. 3A

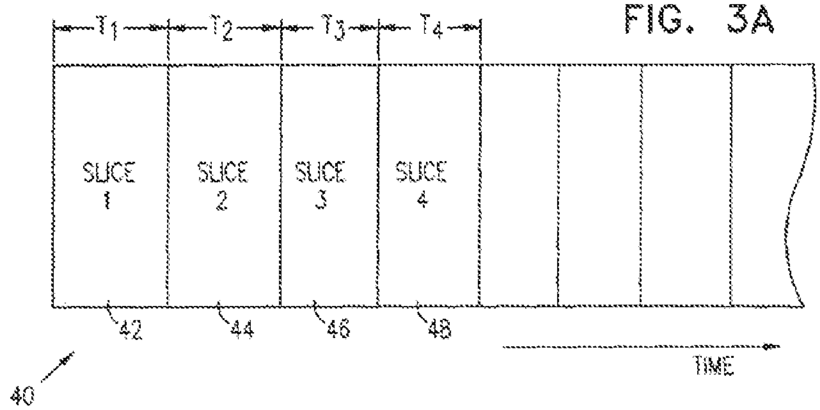


Figure 3A schematically illustrates the structure of broadcast data generated by computer 34, "typically corresponding to a multimedia data sequence." *Id.* at 7:19–23. Data stream 40 comprises a series of data slices 42, 44, 46, 48, etc., with each slice containing a segment of video and/or audio data that corresponds to a respective, successive time interval T_1, T_2, T_3, etc. *Id.* at 7:22–25. Each slice is stored as a corresponding file with a running slice index 1, 2, 3, . . . N, and perhaps also a time stamp that indicates a real time at which the data in the file were recorded or an elapsed time relative to the beginning of the stream. *Id.* at 7:27–32. An index file that comprises a slice ID is uploaded to a server, with the slice ID indicating the index of the file in the data stream that was most recently uploaded. *Id.* at 7:59–64. Each time a new file is uploaded, the slide ID is updated. *Id.* at 7:65–66.

Figure 4 of Carmel is reproduced below.

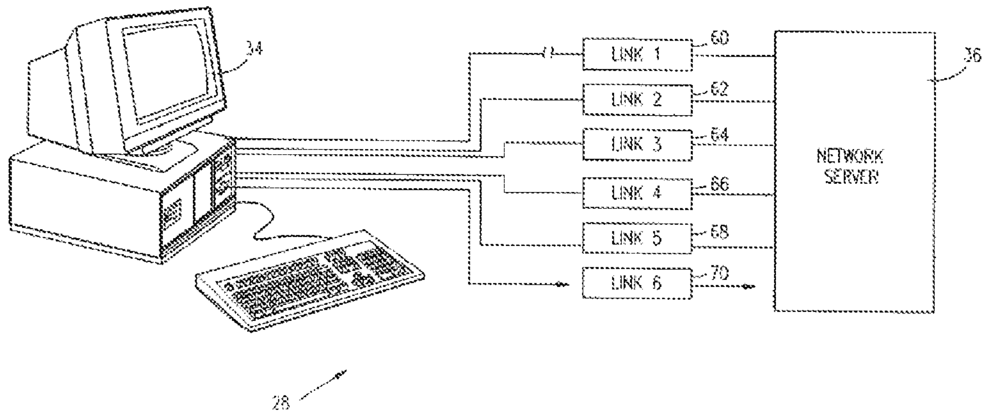


FIG. 4

Figure 4 schematically illustrates communication between computer 34 and server 36 over network 28. *Id.* at 9:10–13. According to Carmel, computer 34 "should preferably ensure that there is sufficient communication bandwidth between the computer and the server." *Id.* at 9:13–17. Accordingly, the computer may open multiple links 60, 62, 64, 66, 68, 70, multiple of which may "operate simultaneously" over a single line or each of which may be "routed differently from the other links" through different lines. *Id.* at 9:17–23.

Computer 34 monitors the rate of data being transmitted over each of the links, and allocates files according to the data rates, perhaps varying file sizes by adjusting slice durations T_1, T_2, T_3, etc. *Id.* at 9:31–37. Carmel notes that "[t]he bandwidth open for transmission between computer 34 and server 36 is effectively roughly equal to a sum of the bandwidths of the plurality of open links." *Id.* at 9:37–39. A similar process is performed when server 36 sends data stream 40 to client computers 30, but, in addition, client computer 30 can read the index file and determine from which slice to begin receiving the data stream. *Id.* at 8:1–9.

1. Anticipation of Claim 10 by Carmel

Petitioner challenges claim 10 as anticipated by Carmel. Pet. 65. Several limitations of claim 10 correspond to limitations recited in independent method claim 1, on which we did not institute review, and Petitioner refers to its analysis of claim 1 for those limitations. *Id.* We have accordingly reviewed Petitioner's analysis for claim 1, *id.* at 50–62, as well as its identification of which elements of claim 10 have counterparts in claim 1, *see id.* at 31–38, and conclude that Petitioner demonstrates, by a preponderance of the evidence, that claim 10 is anticipated by Carmel. The parties dispute only a single, well-defined issue, namely whether Carmel discloses the last limitation of claim 10, which recites "a machine-readable, executable routine containing instructions to cause the server to send media data elements to the user system responsive to said requests, *at a rate more rapid than the rate at which said streaming media is played back by a user*," in light of the Federal Circuit's construction of "rate." Ex. 1001, 14:24–28 (emphasis added); *see* Pet. Remand Br. 1 ("The question on remand is whether Carmel discloses the last limitation of the claim by sending individual requested media data elements (the same 'one or more media data elements' requested in the penultimate limitation) from the server to the user computer 'at a rate more rapid than the rate at which]they are] played back."); PO Remand Br. 2 ("The final element, the 'rate limitation' was the sole point of dispute between the parties regarding claim 10 in this IPR.").

Petitioner identifies two aspects of Carmel that it contends disclose the limitation. First, Petitioner contends that "Carmel discloses that the system in normal operation has a data rate for '**each slice**' that is 'generally equal to **or faster than**' the playback rate." Pet. Remand Br. 2. Second,

Petitioner contends that "Carmel discloses the system can recover from lag (*i.e.*, an interruption in transmission such that transmission is slower than playback) by requesting a lower quality level which will be transmitted faster than playback to catch up." *Id.* In asserting that "Petitioner raised both during the original briefing and they can now be relied upon to meet the new construction," Petitioner cites its Reply. *Id.* (citing Reply, 2, 5–11); *see also* Paper 21, 15:20–16:3 (Petitioner explaining, at oral hearing, responsive nature of argument in Reply).

We have reviewed Petitioner's Reply argument, and agree that Petitioner's discussion of these aspects of Carmel is properly responsive to Patent Owner's argument in its Response that "Carmel does not disclose sending media data elements to a user system responsive to requests therefrom, at a rate more rapid than the rate at which the streaming media is played back by a user." PO Resp. 3 (emphasis omitted). Patent Owner devoted a significant portion of its Response to that argument, and Petitioner was entitled to address it in its Reply—and now to address it further on remand in the context of the Federal Circuit's claim construction. *See Idemitsu Kosan Co., Ltd. v. SFC Co. Ltd.*, 870 F.3d 1376, 1381 ("To the extent Idemitsu suggests that the Board could not reach a counterargument because it was not preemptively addressed by the petition or institution decision, Idemitsu is plainly mistaken."); *Genzyme Therapeutic Prod. Ltd. P'ship v. Biomarin Pharm. Inc.*, 825 F.3d 1360, 1366 (Fed. Cir. 2016) ("There is no requirement, either in the Board's regulations, in the APA, or as a matter of due process, for the institution decision to anticipate and set forth every legal or factual issue that might arise in the course of the trial.").

a. Normal Operation

To support its contention that Carmel teaches sending media data elements at a rate more rapid than the playback rate during "normal operation," Petitioner quotes the following disclosure:

> In some preferred embodiments of the present invention, the transmitting computer and the clients monitor the uploading and downloading of data to and from the server, respectively, in order to determine **the amount of time required to convey each slice and to verify that the slices are conveyed at a sufficient rate**. When the data stream comprises multimedia data, **the data rate should be generally equal to or faster** than the rate at which the data are generated at the transmitting computer.

Pet. Remand Br. 3 (quoting Ex. 1003, 2:51–59) (emphasis by Petitioner). As Petitioner correctly observes, "this passage is describing the 'sufficient rate' during normal streaming operation (uploading and downloading of data to and from the server), not the later-described embodiments using multiple links as one way to compensate for lag or slow connections." *Id.* at 3–4. Petitioner reasons that the last limitation of independent claim 10 is met because Carmel explicitly teaches that such normal operation may use a data rate that is "faster" than the rate at which the data are generated by the transmitting computer. *Id.*

In addition to this intrinsic evidence, Petitioner further supports its contention by pointing to testimony by Patent Owner's expert, Mung Chiang, Ph.D. *See id.* at 5–6. On cross examination, Dr. Chiang explained that "Carmel adjusts the slices so that they are transmitted at about the playback rate." Ex. 1022, 91:10–12. When Petitioner explored the consequences of what Dr. Chiang meant in describing transmission of slices

"at about the playback rate," Dr. Chiang conceded that "[i]f it is transmitted slightly faster than playback rate and then slightly lower, slightly higher, slightly lower, which is what 'about playback rate' means." *Id.* at 92:16–19.

Petitioner argues that, because Carmel teaches that transmission occurs, at least sometimes, at a rate greater than the playback rate, the last limitation of claim 10 is met. Pet. Remand Br. 6 ("This is a dispositive admission—transmitting at least sometimes 'slightly faster' than playback meets the claim limitation of sending 'one or more' requested data elements faster than playback is met."). In advancing that argument, Petitioner relies on *Broadcom Corporation v. Emulex Corporation*, 732 F.3d 1325, 1333 (Fed. Cir. 2013) for the proposition that a claim limitation is met by a device that performs the function "some of the time." *Id.*

Patent Owner does not specifically dispute that Carmel teaches transmission faster than the playback rate, but instead argues that "the flip is also true": "Carmel's sending at about the playback rate necessarily means that it sometimes sends slower than the playback rate." PO Remand Br. 7. Although we agree with Patent Owner that Carmel thus also discloses sometimes transmitting slices at a rate slower than the playback rate, we do not agree with Patent Owner's conclusion that "Carmel cannot teach that *each requested media data element* is sent faster than the playback rate, as required by the Federal Circuit's construction of the rate limitation." *Id.* That conclusion relies on an improper importation of an additional limitation into the claim, namely that *all* requested media data elements must be sent by the server at a rate more rapid than the rate at which the streaming media is played back by a user.

That is, Patent Owner appears to read the word "each" in the Federal Circuit's construction as requiring that *all* media data elements be transmitted faster than the playback rate. As we summarize above, the context in which the Federal Circuit arrived at its construction distinguished from our prior construction of "rate" as corresponding to the "overall rate" of transmission from the server to the user computer, such as might be achieved with multiple links over which data elements are sent to the user system. In referring to "the rate at which each requested data element is transmitted," the Federal Circuit is clearly excluding such an overall rate, as is apparent from its explanation that "[t]he rate limitation in claim 10 therefore refers to the rate at which *requested* media data elements are sent, not the overall rate at which data is transmitted from the server to the user computer." *WAG Acquisition*, 781 F. App'x at 1012. We discern nothing in the Federal Circuit's decision that compels Patent Owner's implicit additional requirement that *all* media data elements be transmitted faster than the playback rate.

For method claims, it is well established that "part-time" satisfaction of a method claim is sufficient to establish anticipation. *See, e.g.*, *Hewlett-Packard Co. v. Mustek Sys. Inc.*, 340 F.3d 1314 (Fed. Cir. 2003) ("a prior art product that sometimes, but not always, embodies a claimed method nonetheless teaches that aspect of the invention"). The same principle analogously applies to the software aspects of apparatus claim 10. That is, the mere fact that the server in Carmel might comprise a machine-readable, executable routine containing instructions to cause the server to send media data elements to the user system at a rate less rapid than the playback rate

does not diminish the fact that it also contains instructions for sending media data elements at a rate more rapid than the playback rate.

We accordingly find that Petitioner sufficiently identifies disclosure in Carmel that meets the last limitation of claim 10 under Carmel's normal operation.

b. Lag Recovery

As an alternative, Petitioner contends that "Carmel further discloses a faster-than-playback transmission rate when it states that '[i]n the event that a lag is detected, steps are taken to **increase the data transmission or reception rate**.'" Pet. Remand Br. 8 (quoting Ex. 1003, 7:39–42) (alterations by Petitioner). In particular, Carmel describes client 30 making an assessment of the rate of data transfer over the link from the server and, if necessary, changing the quality level accordingly. Ex. 1003, 11:9–11. "For example, if the rate is low, such that the time stamps 59 indicate that the slices need to be played as fast or faster than they are being received, the client will preferably select a lower quality level." *Id.* at 11:11–15.

Petitioner reasons that "[b]y the client selecting a lower quality level for the slices, each slice can be individually transmitted faster, and the slices would no longer be 'played as fast or faster than they are being received' thus recovering from lag." Pet. Remand Br. 9. After such a correction, the transmission of the media data elements is faster than the media being played, thereby meeting the claim limitation. *Id.*

As Petitioner notes, we asked at the oral hearing whether the lower-quality slices in Carmel are the same "media data elements" requested earlier in the claim. *See* Paper 21, 13:12–14:16. Petitioner articulates

persuasive reasoning in its remand brief that they are. Specifically, Petitioner observes that "[i]n Carmel, when the *client* determines a new quality level is needed, the *client request* will be for a lower quality level." Pet. Remand Br. 9 (citing Ex. 1003, 11:11–15 ("the client will preferably select a lower quality level")). Because Carmel teaches that "[e]ach time slice in stream 41 includes multimedia data at multiple levels," Ex. 1003, 8:56–57, Petitioner reasons that, although a lower quality level has been requested, "these are still the same time slices (having a predetermined duration and identified by slice ID) that existed since the encoding step." Pet. Remand Br. 9.

The point is illustrated with a specific example, identified by Petitioner, in which each slice has a quality levels described by Carmel as "level #1" and "level #2." *See* Ex. 1003, 8:56–9:5. Carmel explains that "the level #1 slices have smaller data volume than the level #2 slices and can therefore be transmitted over a lower-bandwidth data link, while maintaining the required timing indicated by time stamps 59." *Id.* at 8:57–66. Thus, Petitioner reasons, "when the client determines a new quality level is required, the client will request the lower quality level for that slice ID. Pet. Remand Br. 9–10 (citing Ex. 1003, 11:11–15 ("the client will preferably select a lower quality level")). We agree that Carmel's disclosure accordingly supports Petitioner's assertion that "[t]he server does not ever send new or different files than the ones requested by clients," *id.* at 10; rather each of the clients "chooses . . . the quality level appropriate to the bandwidth of its link on network 28 to server 36," Ex. 1003, 9:6–9.

Patent Owner also addresses this lag-recovery mechanism in its brief. PO Remand Br. 8–9. Patent Owner's assertion that "Carmel's change in

quality only happens after slices are sent at or below the playback rate" is not inconsistent with Petitioner's characterization of this mechanism. *See id.* at 8. But in arguing that "[s]ending slices at or below the playback rate fails to disclose that each requested media data element is sent faster than the playback rate," Patent Owner again implicitly incorporates a requirement, beyond the Federal Circuit's construction, that *every* media data element be transmitted faster than the playback rate. *Id.* at 8–9. For the reasons expressed above, we disagree that such a feature is part of the Federal Circuit's construction of "rate."

We accordingly find that Petitioner sufficiently identifies disclosure in Carmel that meets the last limitation of claim 10 under Carmel's lag-recovery mechanism.

c. Conclusion

As we note above, the parties dispute only whether the last limitation of independent claim 10 is taught by Carmel. Because we find that Petitioner identifies sufficient disclosure in Carmel to meet that limitation, under either normal operation or with a lag-recovery mechanism, we conclude that Petitioner shows, by a preponderance of the evidence, that independent claim 10 is anticipated by Carmel.

2. Anticipation of Claims 11 and 13–18 by Carmel

Petitioner challenges claims 11 and 13–18 as anticipated by Carmel. Pet. 66. In doing so, the Petition makes reference to the analysis provided for corresponding limitations recited in claims that depend from claim 1. *Id.* (referring to analysis for claims 2 and 4–9). These dependent claims recite

that the serial identifiers are sequential (claim 11), that the media is encoded at a variable bit rate (claim 13), that the server is adapted to distribute the streaming media to a plurality of simultaneous users (claim 14), that the server does not maintain a pointer into a buffer established within the server (claim 15), that the operating system comprises a reception protocol "such as TCP" (claim 16), and that the server is adapted to obtain the streaming media from a live source (claim 17) or a disk file (claim 18). Ex. 1001, 14:29–48.

The parties do not separately address the patentability of these dependent claims on remand. We have reviewed the specific Carmel disclosures identified by Petitioner for each of these limitations, and agree that they disclose the respective limitations.[4] *See* Pet. 62–64. Accordingly, we conclude that Petitioner demonstrates, by a preponderance of the evidence, that claims 11 and 13–18 are anticipated by Carmel.

3. *Obviousness of Claim 12 over Carmel and ISO-11172*

Petitioner challenges claim 12, which depends from independent claim 10 and recites that "said media is encoded at a constant bit rate," as unpatentable under 35 U.S.C. § 103(a) over Carmel and ISO-11172. Pet. 68. In doing so, the Petition makes reference to the analysis provided for claim 3, which recites the same limitation but depends from independent claim 1.

---

[4] As noted in our prior Final Written Decision, Patent Owner also does not address the patentability of dependent claims 11, 13, 14, or 16–18 in its Response. *See* Dec. 23. With respect to claim 15, the Federal Circuit held that "[a] reasonable fact finder could find that Carmel does not require use of a pointer for the reasons stated by the Board." *WAG Acquisition*, 781 F. App'x at 1013.

Id. at 67–68. The parties do not separately address the patentability of this claim on remand. For the reasons set forth in our prior Final Written Decision, we agree with Petitioner's reasoning, set forth in its Petition, that, "if the use of a constant bit rate is not inherent in Carmel, a person of ordinary skill in the art would have been motivate to look to ISO-11172 to modify the teachings of Carmel to use a constant bit rate as 'one of the well-known options' of MPEG and 'for the purposes of supporting a wider variety of [media data].'" Dec. 27 (quoting Pet. 67–68).

Accordingly, we conclude that Petitioner demonstrates, by a preponderance of the evidence, that dependent claim 12 is unpatentable under 35 U.S.C. § 103(a) over Carmel and ISO-11172.

III. CONCLUSION

In summary, we make the following conclusions.[5]

Claims	35 U.S.C. §	Reference(s)	Claims Shown Unpatentable	Claims Not Shown Unpatentable
10, 11, 13–18	102(a), 102(e)	Carmel	10, 11, 13–18	
12	103(a)	Carmel and ISO-11172	12	

[5] Should Patent Owner wish to pursue amendment of the challenged claims in a reissue or reexamination proceeding subsequent to the issuance of this decision, we draw Patent Owner's attention to the April 2019 *Notice Regarding Options for Amendments by Patent Owner Through Reissue or Reexamination During a Pending AIA Trial Proceeding*. *See* 84 Fed. Reg. 16,654 (Apr. 22, 2019). If Patent Owner chooses to file a reissue application or a request for reexamination of the challenged patent, we remind Patent Owner of its continuing obligation to notify the Board of any such related matters in updated mandatory notices. *See* 37 C.F.R. § 42.8(a)(3), (b)(2).

IV. ORDER

It is

ORDERED that, based on a preponderance of the evidence, claims 10–18 of U.S. Patent No. 8,122,141 B2 are held to be unpatentable; and

FURTHER ORDERED that, because this is a final written decision, parties to this proceeding seeking judicial review of our decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

PETITIONER

Frank Gasparo
Jonathan Falkler
VENABLE, LLP
fmgasparo@venable.com
JLFalkler@venable.com

PATENT OWNER:

Ronald Abramson
Ernest Buff
LEWIS, BAACH, KAUFMANN, MIDDLEMISS, PLLC
Ronald.abramson@lbkmlaw.com
ebuff@edbuff.com

Exhibit 14
to
Ex Parte Reexamination
of
U.S. Patent No.
8,327,011

Notice of Allowability	**Application No.** 13/374,942	**Applicant(s)** PRICE, HAROLD EDWARD	
	Examiner MARSHALL MCLEOD	**Art Unit** 2454	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *01/24/2012*.
2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.
3. ☒ The allowed claim(s) is/are *1-4*.
4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE “MAILING DATE” of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER’S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
6. ☐ CORRECTED DRAWINGS (as “replacement sheets”) must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson’s Patent Drawing Review (PTO-948) attached
   1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner’s Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
7. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner’s comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date 01/24/2012
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☐ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

/Joseph E. Avellino/
Supervisory Patent Examiner, Art Unit 2454

DETAILED ACTION

Reasons for Allowance

1. The following is an examiner's statement of reasons for allowance: Claims 1-4 are allowable over the prior art as the prior art did not teach nor suggest independently or in combination: A media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number and instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer and instructions to cause the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received. For these reasons, in conjunction with the limitations of the independent claims, puts this case in condition for allowance. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MARSHALL MCLEOD whose telephone number is (571)270-3808. The examiner can normally be reached on Monday - Thursday 6:30 a.m-4:00 p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, JOSEPH AVELLINO can be reached on (571) 272-3905. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/M. M./
Examiner, Art Unit 2454
5/17/2012

Trials@uspto.gov
571-272-7822

Paper 7
Date Entered: December 12, 2016

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

WEBPOWER, INC.,
Petitioner,

v.

WAG ACQUISITION, LLC,
Patent Owner.

Case IPR2016-01161
Patent 8,327,011 B2

Before GLENN J. PERRY, TREVOR M. JEFFERSON, and BRIAN J. McNAMARA, *Administrative Patent Judges.*

McNAMARA, *Administrative Patent Judge.*

DECISION
Denying Institution of *Inter Partes* Review
37 C.F.R. § 42.108

new data element is received into the FIFO buffer it is immediately sent to the user computer so as to keep the user computer buffer full. *Id.* at 9:22–25. The process repeats for the entire time the audio/video material is played. *Id.* at 9:26–27. The audio/video material is transmitted from the server more rapidly than it is played, allowing the user buffer to build up and to be restored if it is diminished by interruptions. *Id.* at 9:28–44.

ILLUSTRATIVE CLAIM

Claim 1 is illustrative:

> 1. A media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number, comprising
>
> a processor;
>
> a memory;
>
> a connection to the network; and
>
> media player software comprising instructions to cause the media player to request from the media source a predetermined number of media data elements;
>
> instructions to cause the media player to receive media data elements sent to the media player by the media source and store the media data elements in the memory;
>
> instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer;
>
> instructions to cause the media player to play media data elements sequentially from the player buffer; and
>
> instructions to cause the media player to transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data

elements in the player buffer until the last media data element comprising the program has been received.

ART CITED IN PETITIONER'S CHALLENGES

Petitioner cites the following references in its challenges to patentability:

Title	Designation	Exhibit No.
U.S. Patent No. 5,822,524, issued Oct. 13, 1998	Chen	Ex. 1002
U.S. Patent No. 7,065,342 B1, issued June 20, 2006	Rolf	Ex. 1003

CHALLENGES ASSERTED IN PETITION

Claim(s)	Statutory Basis	Challenge
1, 2, 4	35 U.S.C. § 102	Anticipation by Chen
3	35 U.S.C. § 103(a)	Obvious over Chen in view of Rolf

CLAIM CONSTRUCTION

We interpret claims of an unexpired patent using the broadest reasonable construction in light of the specification of the patent in which they appear. *See* 37 C.F.R. § 42.100(b); *In re Cuozzo Speed Techs., LLC*, 793 F.3d 1268, 1278 (Fed. Cir. 2015) *cert. granted sub nom. Cuozzo Speed Techs. LLC v. Lee*, 136 S. Ct. 890 (mem.) (2016). In applying a broadest reasonable construction, claim terms generally are given their ordinary and customary meaning, as would be understood by one of ordinary skill in the art in the context of the entire disclosure. *See In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007). Any special definition for a claim term must be set forth in the specification with reasonable clarity,

deliberateness, and precision. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994).

Petitioner proposes that we construe the claim terms to have their ordinary and customary meaning. Pet. 12. Patent Owner agrees that the claims can be construed in accordance with their plain and ordinary meaning. Prelim. Resp. 4. However Patent Owner argues that, because it includes the word "sequential," we should construe the phrase "to repeat transmitting the requests to the media source for *sequential* media data elements so as to maintain the pre-determined number of data elements in the player buffer until the last media data element comprising the program has been received" to mean that "the client repeatedly requests the elements sequentially (meaning in the sequence in which they are to be played)." *Id.* (citing Ex. 1001, 8:63–66, which reads "As data is played out, the next sequential data elements are requested from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user's buffer.").

The '011 Patent states that "the invention provides a system for distributing via the Internet streaming media composed of a plurality of time-sequenced data elements." Ex. 1001, 4:43–45. The Specification also states:

> The invention presumes the existence of a data communications transport mechanism, such as the TCP protocol, for the reliable delivery of data in an ordered sequence from the source of the media data to the server, or from the server to the media player software of the user computer. Thus, the delivery of data in the proper sequence is outside the scope of this invention.

Id. at 4:61–67.

comprises instructions "to repeat transmitting the requests to the media source for sequential media data elements."

Patent Owner argues that Petitioner has not established that Chen discloses the portion of the limitation that recites "to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received." Prelim. Resp. 5–6, 12–15.

The testimony of Petitioner's expert, Dr. Polish, appears to be consistent with Patent Owner's position. Dr. Polish identifies two processes in Chen for keeping track of the last packet transmitted, the first being the detection of lost packets. Polish Decl. ¶ 40. As discussed above, lost packets are requested individually by number, as opposed to sequentially. Ex. 1001, 10:42–50.

In the second of the two Chen processes identified by Dr. Polish "the server paces transmission in the Normal mode such that the client agent is not required to send periodic feedback to the server control." *Id.* (citing Ex. 1002, 6:32–39). The absence of the need for periodic feedback suggests that there is no need in Chen for the client "to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received." To that extent Dr. Polish's testimony is consistent with Patent Owner's position that Chen does not disclose repeating requests to transmit a predetermined number of media data elements. We further note Dr. Polish's testimony that

"Chen's server necessarily tracks the last element sent so as to be able to send the next sequential element without client feedback." Polish Decl. ¶ 40. This is significant because claim 1 recites transmitting requests to the server. Patent Owner argues that Petitioner's focus on activity in the server side of Chen is misplaced and irrelevant to the subject matter of claim 1, which is drawn to instructions in the media player software. Prelim. Resp. 13.

Patent Owner also argues that the Petition does not address the claim term "repeat." Prelim. Resp. 12. Patent Owner contends that subject matter of claim 1, in which the media player software in the client repeats requests to the server for sequential server elements, is different from Chen because Chen operates in a "push" mode with the server sending data in a NORMAL or RUSH mode, instead of responding to requests for individual packets by their serial identifiers. *Id.* at 12–13. According to Patent Owner, the claim language requires that the "repeat" element must be in the form of requests from the client side for transmission of media elements by the server. *Id.* at 12. Patent Owner argues that, although Chen discloses the client monitoring the amount of data in its client packet buffer, the client acts to issue commands requiring the server to change its transmission mode to speed up, slow down or halt data transmissions, not to request individual packets. *Id.* at 14.

Claim 1 recites "instructions to cause the media player to transmit to the media source a request to send one or more media data elements, each identified by serial number" and "to repeat transmitting the requests to the media source for sequential media data elements." Thus, the repeated requests must be for one or more data elements each identified by serial

number. As discussed above, Chen does not disclose such requests, but only a command that triggers the server to adjust the speed of transmission of data elements. In Chen, transmission scheduling is controlled by the transmission scheduler in the server control. Ex. 1002, 6:48–7:38, 9:6–7, 21–29. The client sets the values of the transmit mode and the frame time via command packets. *Id.* at 10:39–42. Notwithstanding that the data elements in Chen are identified by sequence and frame numbers, or even that Chen and the system in the '011 Patent cause the data packets to be transmitted in the same sequence, the mode commands in Chen do not identify the sequential media data elements to be transmitted by a serial number—the requests in Chen simply signal the server to adjust the speed of data elements it transmits.

For the reasons discussed above, Petitioner has not demonstrated that Chen discloses all the elements of claim 1. For this reason, Petitioner has not demonstrated a reasonable likelihood that it will prevail in its challenge to claim 1 of the '011 Patent as anticipated by Chen. Claims 2 and 4 depend from claim 1 and, for the same reasons, Petitioner has not demonstrated that these claims are anticipated by Chen.

Claim 3 As Obvious Over Chen and Rolf

Claim 3 recites "[a] wireless phone adapted to function as a media player in accordance with claim 1." Petitioner argues that "[t]he system architecture and client machine (20) described in Chen could easily apply to a media player on a wireless phone." Pet. 54. As discussed above, we have determined that Chen does not disclose the features of claim 1. Petitioner does not argue any other disclosure in Rolf discloses the features of claim 1 we find are not disclosed in Chen. Thus, Petitioner has not demonstrated

that claim 3 would have been obvious to one of ordinary skill under the provisions of 35 U.S.C. 103(a).

SUMMARY

For the reasons discussed above, Petitioner has not demonstrated a reasonable likelihood that it will prevail on its challenges to the patentability of claims 1–4 of the '011 Patent and we decline to institute *inter partes* review.

ORDER

In consideration of the foregoing, it is hereby:

ORDERED that *inter partes* review is not instituted.

Trials@uspto.gov
571-272-7822

Paper 11
Date Entered: February 27, 2017

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

I.M.L. SLU,
Petitioner,

v.

WAG ACQUISITION, LLC
Patent Owner.

Case IPR2016-01655
Patent 8,327,011 B2

Before TREVOR M. JEFFERSON, BRIAN J. McNAMARA, and PATRICK BOUCHER, *Administrative Patent Judges.*

McNAMARA, *Administrative Patent Judge.*

DECISION
Denying Institution of *Inter Partes* Review
37 C.F.R. § 42.108

WebPower had not demonstrated it was likely to prevail in its challenge to claims 1, 2, and 4 as anticipated by Chen. *WebPower, Inc. v. WAG Acquisition, LLC*, Case IPR2016-01161, slip op. (PTAB Dec. 12, 2016). Patent Owner contends that, although Petitioner seeks to apply a new spin to old facts, a side-by-side comparison of the Petition against the petition WebPower filed in IPR2016-01161 makes clear that Petitioner's anticipation arguments are substantially similar to the arguments that WebPower advanced and that we found unpersuasive. Prelim. Resp. 1–2.

In *WebPower-1161*, we determined that Chen discloses some of the elements of claim 1, but not others. *WebPower*-1161 Dec. Denying Inst. 11–20. We do not repeat the entire analysis here. However, in *WebPower*-1161, we found that Chen does not disclose instructions that cause the media player to request from the media source a predetermined number of media data elements, as recited in claim 1. *Id.* at 13–15. We noted:

> Chen operates on a different principle from that articulated in the '011 Patent. Chen employs a normal data transmission rate, based on number of factors, including the number of frames that can be stored in the client packet buffer, and causes the server to adjust its data transmission rate in response to the actual amount of data in the client packet buffer. . . Chen maintains an acceptable range of numbers of packets in the client packet buffer, although this range is predetermined by the high and low Water Marks. The client's issuance of commands causing the server to alter the data transmission rate until specific conditions defined by the high and low Water Marks are met could cause transmission of a predetermined number of packets that could be calculated based on the characteristics of the packets and the transmission rate. However, Chen does not disclose explicitly requesting a predetermined number of packets. Instead, Chen discloses transmission in the NORMAL mode most of the time, stating that "[t]ransmission occurs very efficiently in this NORMAL mode because no need exists for the client agent (30)

> to send periodic feedback to the server control (1)." Ex. 1002, 6:32–39.

Id. at 15. Petitioner's arguments in this proceeding are almost the same as those advanced in *WebPower-1161*. Pet. 27–28. For the same reasons as those reproduced above from *WebPower-1161*, we again find that Chen does not disclose this element.

The final limitation of claim 1 recites "instructions to cause the player to transmit to the media source a request to send one or more media data elements each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer . . ." In *WebPower-1161*, we stated:

> [R]epeated requests must be for one or more data elements each identified by serial number. As discussed above, Chen does not disclose such requests, but only a command that triggers the server to adjust the speed of transmission of data elements. In Chen, transmission scheduling is controlled by the transmission scheduler in the server control. Ex. 1002, 6:48–7:38, 9:6–7, 21–29. The client sets the values of the transmit mode and the frame time via command packets. Id. at 10:39–42. Notwithstanding that the data elements in Chen are identified by sequence and frame numbers, or even that Chen and the system in the '011 Patent cause the data packets to be transmitted in the same sequence, the mode commands in Chen do not identify the sequential media data elements to be transmitted by a serial number—the requests in Chen simply signal the server to adjust the speed of data elements it transmits.

Id. at 20. Thus, in Chen there are no repeated requests for data elements, as recited in claim 1 of the '011 Patent—only requests to set the speed of data transmission. Petitioner also advances an argument focused on requests for retransmission of lost packets. Pet. 31–33. As to requests for retransmission

of lost packets, we noted in *WebPower-1161* and again note that these are requests for retransmission of individual lost packets and not "requests to the media source for sequential media data elements" as recited in claim 1. Ex. 1001, 10:42–50. Thus, we again find that this limitation is not disclosed in Chen.

For the reasons discussed above, we are not persuaded that Petitioner has demonstrated a reasonable likelihood it will prevail in its challenge to independent claim 1, and dependent claims 2 and 4, as anticipated by Chen.

Claim 2 As Obvious Over Chen In View of Chen File History

Claim 2 depends from claim 1 and recites a computer adapted to function as a media player in accordance with claim 1. Petitioner cites Chen File History to the extent that we determine Chen does not disclose the feature that the media player, i.e. the client agent, receives media data elements at a rate more rapid that the rate at which the media data elements are to be played out." Pet. 56–57. Petitioner does not identify anything in Chen File History that discloses the elements of claim 1 of the '011 Patent that we find are not disclosed in Chen. Thus, we are not persuaded that Petitioner has demonstrated a reasonable likelihood it will succeed in its challenge to claim 2 as obvious over Chen in view of Chen File History.

Claim 3 As Obvious Over Chen In View of Galensky and White

Claim 3 depends from claim 1 and recites a wireless phone adapted to function as the media player. Petitioner acknowledges that Chen does not disclose a media player on a wireless phone and cites Galensky and/or White for the proposition that it would have been obvious to try Chen's media player on the wireless devices and phones of Galensky and White. Pet. 59. Petitioner does not identify any disclosure in Galensky or White of the

elements of claim 1 that we find are not disclosed in Chen. Thus, we are not persuaded that Petitioner has demonstrated a reasonable likelihood it will prevail in its challenge to claim 3 as obvious over Chen in view of Galensky and White.

SUMMARY

For the reasons discussed above, we are not persuaded that Petitioner has demonstrated a reasonable likelihood that it will succeed on its challenges to patentability of claim 1–4 of the '011 Patent.

ORDER

In consideration of the above, we decline to institute *inter partes* review of claim 1–4 of the '011 Patent.

PETITIONER:

David Yohannan
Beth Jacob
KELLEY DRYE & WARREN LLP
dyohannan@kelleydrye.com
bjacob@kelleydrye.com

PATENT OWNER:

Ronald Abramson
LEWIS BAACH PLLC
ronald.abramson@lewisbaach.com

Ernest Buff
ERNEST D. BUFF & ASSOCIATES, LLC
ebuff@edbuff.com

Exhibit 15
to
Ex Parte Reexamination
of
U.S. Patent No.
8,327,011

UNITED STATES DISTRICT COURT
DISTRICT OF NEW JERSEY

WAG ACQUISITION, L.L.C., Plaintiff, v. Gattyán Group S.à r.l., *et al.*, Defendants.	Case No.: 14-cv-02832 (ES) (MAH) JURY TRIAL DEMANDED

PLAINTIFF'S OPENING CLAIM CONSTRUCTION BRIEF

8. *Item 12: The "sending rate" limitation*

A. *"sending rate"*
B. *"send[] … at … rate"*

WAG's Proposed Construction	Duodecad's Proposed Construction
Plain and Ordinary Meaning *A:* The rate at which the media is sent. *B:* The rate at which the media is sent.	"number of bits per unit time the server transmits over the transport mechanism"

Yet again, Duodecad espouses the constant bit rate argument.

The term "sending rate" appears in claim 2 of the '839 Patent, and claims 1, 3, 8, 9, and 15 of the '611 Patent. The other term appears in claim 1 of the '141 Patent and claims 1, 3, 8, 9, and 15 of the '611 Patent. The plain meaning of these terms is simply a rate at which something is sent. In the context of the Asserted Claims, that "something" is media, and thus "sending rate" and "send[] … at … rate" simply mean the rate at which media is sent. Nothing in the claims or specifications of the Asserted Patents suggests a contrary interpretation.

Duodecad seeks to narrow these "sending rate" terms to a specific type of rate: the "number of bits per unit time delivered to the transport mechanism." This wholesale rewriting of the claims not only introduces unneeded complexity but is also contradicted by the plain language of the claims themselves and the specifications of the Asserted Patents.

Claim 2 of the '839 Patent recites sending media data elements "at a sending rate more rapid than the playback rate." '839 Patent, 16:26-29. Similarly, claim 1 of the '611 Patent recites sending media data elements "at an initial rate more rapid than the playback rate." '611 Patent, 15:63-65. The "sending rate" is thus compared to the "playback rate." But, as noted above, it is improper to limit "playback rate" to a bit rate, and thus it would also be improper to limit "sending rate" to a "bit rate."

Duodecad's construction is also improper in that it converts "sending" to "delivered to the transport mechanism." While it is true that the Asserted Patents disclose that "[t]he server buffer 14 'sends' data by delivering it to the transport mechanism" (*e.g.*, '611 Patent, 8:14-15), such disclosure provides no basis to convert the simple term "sending" into such a complicated, technical phrase. Moreover, the term "delivered to the transport mechanism" seeks to explicitly limit the claims to a single disclosed embodiment. But nothing in the Asserted Patents, the Asserted Claims, or their related file histories suggests that the Asserted Claims must be limited to this disclosed embodiment. *See Blackbird Tech*, 895 F.3d at 1377; *In re Hiniker Co.*, 150 F.3d 1362, 1368 (Fed. Cir. 1998) ("Although operational characteristics of an apparatus may be apparent from the specification, we will not read such characteristics into the claims when they cannot be fairly connected to the structure recited in the claims").

9. Item 13: "serial number / identifier"

WAG's Proposed Construction	Duodecad's Proposed Construction
Plain and Ordinary Meaning Identifiers that correspond to a sequence.	"a consecutive number that identifies a respective media data element"

The term "serial identifier" appears in claims 5 and 8 (through their dependency on claim 1) and claim 24 of the '141 Patent, while the term "serial number" appears in claim 1 of the '011 Patent.

Claim 1 of the '011 Patent recites a "program comprising media data elements," in which "each of the media data elements is associated with a serial number." '011 Patent, 13:14-17. A "serial number" is "[a] number that is one of a series and is used for identification, as of a machine, weapon, or motor vehicle." https://ahdictionary.com/word/search.html?q=serial+number. Nothing in the plain meaning of

"serial number" requires that it be "consecutive" as Duodecad would have it, though the term embraces both concepts. Indeed, notwithstanding the term "number" present in this term, nothing in the common understanding of "serial number" requires that it even be made wholly and exclusively from numbers. A "vehicle identification **number**," for example, is not formed purely from numbers; so, too, for a serial number. In the context of claim 1 of the '011 Patent, a POSITA would recognize that this term simply means an identifier that corresponds to a sequence in connection with the media data elements.

Similarly, claim 1 of the '141 Patent recites "streaming media comprising a plurality of sequential media data elements," and then, in relation to these media data elements, "corresponding [] serial identifiers." *See* '141 Patent, 13:25-30. Like a "serial number," a POSITA would understand that a "serial identifier" simply identifies a corresponding media data element in the sequence (that is, series) of media data elements.

Moreover, a POSITA would further recognize that such serial numbers or identifiers need not be "consecutive." Indeed, claim 2 of the '141 Patent makes this clear, as it further limits "serial identifiers" to being "sequential." *See* '141 Patent, 13:45-46. Under the claim differentiation doctrine, a narrowing limitation implies that the term has broader scope in the parent claim, and thus a "serial identifier" need not necessarily be consecutive or sequential. *See Clearstream Wastewater Sys., Inc. v. Hydro-Action, Inc.*, 206 F.3d 1440, 1446 (Fed. Cir. 2000) ("Under the doctrine of claim differentiation, it is presumed that different words used in different claims result in a difference in meaning and scope for each of the claims" which "prevents the narrowing of broad claims by reading into them the limitations of narrower claims.").

10. *Item 14: "user system being assumed to have a user buffer for receiving media data and facilities to play back the streaming media"*

WAG's Proposed Construction	Duodecad's Proposed Construction
Plain and Ordinary Meaning The server is programmed on the assumption that the receiving media player has a user buffer for receiving media data and facilities to play back the streaming media at [a/the] playback rate for viewing or listening by said at least one user.	"server software does not check that a user system has a user buffer for receiving media data and facilities to play back the streaming media"

This term appears in claims 1, 3, 8, 9, and 15 of the '611 Patent, and in claim 2 (via dependence from claim 1) of the '839 Patent. In each case, this limitation appears in the preambles of the respective claims, reciting the intended use of the claimed invention. In this context, it is clear that this limitation is simply reciting aspects that will provide background for the body of the claim. That is, that the limitations recited in the body have, as background, the underlying assumption stated in the preamble that the user system has a user buffer for receiving media data and facilities to play back the streaming media, *e.g.*, a media player.

Whether the server software checks this assumption is immaterial. Nothing in specifications of the '611 and '839 Patents contradict this understanding. Notably, there is *no disclosure* (let alone claim limitation) that the "server software *does not check* that a user system has a user buffer for receiving media data and facilities to play back the streaming media," as Duodecad would read into the claim.

According to the specification, "[i]nitially, the user buffer manager 22 requests the server 12 to send media data elements to start the playback stream, such as by selecting a radio station from a list. The server 12 responds by sending data elements to the user computer 18 at higher than the playback rate, until the entire FIFO buffer 14 has been sent to the user computer." '611 Patent, 9:18-23; '839 Patent, 9:18-23. To the extent that, in one disclosed embodiment, it is the

Exhibit 16
to
Ex Parte Reexamination
of
U.S. Patent No.
8,327,011

**UNITED STATES DISTRICT COURT
DISTRICT OF NEW JERSEY**

WAG ACQUISITION, L.L.C., Plaintiff, v. Gattyán Group S.à r.l., *et al.*, Defendants.	Case No.: 14-cv-02832 (ES) (MAH) JURY TRIAL DEMANDED

PLAINTIFF'S RESPONSIVE CLAIM CONSTRUCTION BRIEF

Duodecad's construction, and the playback that the user sees, that is, the application layer "playback rate," thus further highlighting Duodecad's faulty construction.

The Court should adopt the plain and ordinary meanings of these simple terms.

8. Item 12: The "sending rate" limitation

A. "sending rate"
B. "send[] ... at ... rate"

Duodecad's statement that "Defendant's construction amplifies how rates are measured in the data transfer context. In contrast, WAG's construction adds nothing," is actually a good point if one understands what "amplifies" means: unduly limits. Duodecad Opening Br. at 25. And, of course, this quote shows that Duodecad is seeking to again inject transport layer technicalities into claims directed to application layer functionality.

In justifying why the technicalities of "bits per unit time" must be injected into this term, Duodecad states that "[d]efining 'rate' in an objective measure of bits over time will assist the factfinder in making a rate assessment when comparing sending rates to playback rates." *Id.* at 24. But Duodecad never explains why a fact finder cannot just compare the sending rate to the playback rate, rather than being forced to analyze transport layer details of bitrates. Nor does Duodecad explain away the disclosure in the specification that "[t]he server buffer 14 'sends' data by delivering it to the transport mechanism," indicating that the sending rate is something other than (or at least a superset of) the underlying transport layer bit rate. '611 patent, 8:14-15. As for Duodecad's citation to arguments made in an IPR, the quote relied upon compares sending rate to playback rate, with an ancillary note that the underlying transport mechanism must be able to support transmissions faster than the playback rate; nothing about this forces the conclusion that the sending rate must be a bit rate.

In short, Duodecad has pointed to no evidence in the intrinsic record to limit the sending rate to only a bit rate and thus its construction should be ignored in favor of the plain and ordinary meaning of these terms.

9. Item 13: "serial number / identifier"

Nothing in the plain language of these two terms requires them to be either consecutive or limited to numerals, as argued in WAG's opening brief. *See* D.I. 274 at 22-23. Serial items indicate a series and requiring them to be consecutive would render superfluous claim 2 of the '141 patent. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1315 (Fed. Cir. 2005) (*en banc*) ("[T]he presence of a dependent claim that adds a particular limitation gives rise to a presumption that the limitation in question is not present in the independent claim.").

Just because there is a "series" of serial numbers / identifiers does not mean that they must be consecutive. Non-consecutive numbers can represent a series.

Duodecad's references to WAG's arguments made in IPRs are inapposite. WAG did not argue that systems using pre-existing identifiers were excluded from the "claims of the Asserted Patents." Duodecad Opening Br. at 25 (citing IPR 2015-01037, Paper 6 at 30 (D.I. 273-14)). This argument was not related to all of the claims of the Asserted Patents, but was directed to claims 19-23 of the '141 Patent, which specifically recite a routine to "serially identify" sequential data elements, *i.e.*, the act of identifying them. These claims are no longer asserted in this case. IPR 2015-01037, Paper 6 (D.I. 273-14) at 30 (heading above Duodecad's quote, ending in "Claims 19-23."). WAG also never argued that the "identifier" in claims 1 and 24 of the '141 Patent is a "serial number" as Duodecad argues. *See* Duodecad Opening Br. at 25 (citing IPR 2015-01037, Paper 6 (D.I. 273-14) at 40-42). The asserted prior art included "serial numbers" and WAG was distinguishing these from the identifiers of the claims using the terminology of the asserted art itself.

Moreover, Duodecad tries to conflate the fact that the media data elements themselves are "sequential" with some requirements that the identifiers themselves are consecutive. The series of the identifiers / numbers *corresponds* to a sequence. That sequence is the multimedia data itself. All multimedia data is sequential because music and video has an order, such that you perceive the notes or images in the proper order and not scrambled. The media data elements are sequential. The identifiers / numbers are in a series that identifies that sequence, but the identifiers / numbers are not themselves necessarily in a sequence or consecutive, they are merely in a defined series so that they correspond to the sequence of the multimedia data elements.

Finally, Duodecad's cite to the Oxford Dictionary (Duodecad Opening Br. at 26 (citing COMMON000169814, D.I. 273-19)) shows that WAG's definition is correct. An item that shows a position in a series is perfectly in-line with identifiers that correspond to a sequence. This definition in no way supports the more limiting "consecutive." It suggests just the opposite -- that the identifiers / numbers need <u>not</u> be consecutive.

10. Item 14: "user system being assumed to have a user buffer for receiving media data and facilities to play back the streaming media"

Duodecad argues, based on dictionary definitions, that this clause from the claim preambles must be a specific, and negative limitation on the steps that the overall system takes. It is not so. As argued in WAG's opening brief (D.I. 274 at 24-25), this term merely means that the system is constructed under the assumption that the user system has a buffer with the stated properties. An actual check for the buffer, whether direct or indirect, is neither included or excluded from the claim itself and, as argued in WAG's opening brief, some disclosures actually include a check for a buffer. *Id.*

11. Item 15: "from a server assumed to be capable of sending streaming media elements at a rate more rapid than the rate at which said streaming media is played back by a user"

As with Item 14 above, there is no specific negative requirement in this language.

12. Item 17: "about the playback rate"

The Parties nearly agree with the construction of this term, except that Duodecad, again, insists upon inserting the further limitation that "'rate' cannot be met/measured using aggregates or averages," which is pure surplusage that finds no support in the claim language itself or in the specification. As discussed above in relation to Item 2, WAG's prior arguments regarding averages are not applicable. One went to "long term averages" and the other was not actually an argument about averages, but rather about an "aggregate."

The Court should thus adopt the plain meaning for this term, *e.g.*, at approximately the playback rate, which is the rate at which the media will be played out in a normal rendition. *See Modine Mfg. Co. v. U.S. Int'l Trade Comm'n*, 75 F.3d 1545, 1554 (Fed. Cir. 1996), *abrogated on other grounds by Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.*, 234 F.3d 558 (Fed. Cir. 2000) (terms such "as 'about' must be given reasonable scope … [and] viewed by the decisionmaker as they would be understood by persons experienced in the field of the invention").

13. Item 19: The "maintain" limitation

A. "media player software comprising…instructions to implement a player buffer manager…operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer"

B. "software being programmed to cause the media player to maintain a record of the identifier of the last data element that has been received"

Exhibit 17
to
Ex Parte Reexamination
of
U.S. Patent No.
8,327,011

HONORABLE BARBARA J. ROTHSTEIN

UNITED STATES DISTRICT COURT
WESTERN DISTRICT OF WASHINGTON
AT SEATTLE

WAG ACQUISITION, LLC, Plaintiff, v. FLYING CROCODILE, INC., d/b/a FCI, INC., et al. Defendants.	Case No. 2:19-cv-01278-BJR **PLAINTIFF'S OPENING CLAIM CONSTRUCTION BRIEF**

CORR CRONIN LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

construction is simply confusing in the context of VBR-encoded data, in which the "number of bits per unit time" being "played out" will change during the course of a program.

8. Item 12: The "sending rate" limitation

Yet again, FCI espouses the constant bit rate argument.

The term "sending rate" appears in claim 2 of the '839 Patent, and claims 1, 3, 8, 9, and 15 of the '611 Patent. The other term appears in claim 1 of the '141 Patent and claims 1, 3, 8, 9, and 15 of the '611 Patent. The plain meaning of these terms is simply a rate at which something is sent. In the context of the Asserted Claims, that "something" is media, and thus "sending rate" and "send[] … at … rate" simply mean the rate at which media is sent. Nothing in the claims or specifications of the Asserted Patents suggests a contrary interpretation.

FCI seeks to narrow these "sending rate" terms to a specific type of rate: the "number of bits per unit time delivered to the transport mechanism." This wholesale rewriting of the claims not only introduces unneeded complexity but is also contradicted by the plain language of the claims themselves and the specifications of the Asserted Patents.

Claim 2 of the '839 Patent recites sending media data elements "at a sending rate more rapid than the playback rate." '839 Patent, 16:26-29. Similarly, claim 1 of the '611 Patent recites sending media data elements "at an initial rate more rapid than the playback rate." '611 Patent, 15:63-65. The "sending rate" is thus compared to the "playback rate." But, as noted above, it is improper to limit "playback rate" to a bit rate, and thus it would also be improper to limit "sending rate" to a "bit rate."

FCI's construction is also improper in that it converts "sending" to "delivered to the transport mechanism." While it is true that the Asserted Patents disclose that "[t]he server buffer 14 'sends' data by delivering it to the transport mechanism" (*e.g.*, '611 Patent, 8:14-15), such disclosure provides no basis to convert the simple term "sending" into such a complicated,

CORR CRONIN LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

technical phrase. Moreover, the term "delivered to the transport mechanism" seeks to explicitly limit the claims to a single disclosed embodiment. But nothing in the Asserted Patents, the Asserted Claims, or their related file histories suggests that the Asserted Claims must be limited to this disclosed embodiment. *See Blackbird Tech*, 895 F.3d at 1377; *In re Hiniker Co.*, 150 F.3d 1362, 1368 (Fed. Cir. 1998) ("Although operational characteristics of an apparatus may be apparent from the specification, we will not read such characteristics into the claims when they cannot be fairly connected to the structure recited in the claims").

9. Item 13: "serial number / identifier"

The term "serial identifier" appears in claims 5 and 8 (through their dependency on claim 1) and claim 24 of the '141 Patent, while the term "serial number" appears in claim 1 of the '011 Patent.

Claim 1 of the '011 Patent recites a "program comprising media data elements," in which "each of the media data elements is associated with a serial number." '011 Patent, 13:14-17. A "serial number" is "[a] number that is one of a series and is used for identification, as of a machine, weapon, or motor vehicle." https://ahdictionary.com/word/search.html?q=serial+number. Nothing in the plain meaning of "serial number" requires that it be "consecutive" as FCI would have it, though the term embraces both concepts. Indeed, notwithstanding the term "number" present in this term, nothing in the common understanding of "serial number" requires that it even be made wholly and exclusively from numbers. A "vehicle identification **number**," for example, is not formed purely from numbers; so, too, for a serial number. In the context of claim 1 of the '011 Patent, a POSITA would recognize that this term simply means an identifier that corresponds to a sequence in connection with the media data elements.

Similarly, claim 1 of the '141 Patent recites "streaming media comprising a plurality of sequential media data elements," and then, in relation to these media data elements,

CORR CRONIN LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

"corresponding [] serial identifiers." *See* '141 Patent, 13:25-30. Like a "serial number," a POSITA would understand that a "serial identifier" simply identifies a corresponding media data element in the sequence (that is, series) of media data elements.

Moreover, a POSITA would further recognize that such serial numbers or identifiers need not be "consecutive." Indeed, claim 2 of the '141 Patent makes this clear, as it further limits "serial identifiers" to being "sequential." *See* '141 Patent, 13:45-46. Under the claim differentiation doctrine, a narrowing limitation implies that the term has broader scope in the parent claim, and thus a "serial identifier" need not necessarily be consecutive or sequential. *See Clearstream Wastewater Sys., Inc. v. Hydro-Action, Inc.*, 206 F.3d 1440, 1446 (Fed. Cir. 2000) ("Under the doctrine of claim differentiation, it is presumed that different words used in different claims result in a difference in meaning and scope for each of the claims" which "prevents the narrowing of broad claims by reading into them the limitations of narrower claims.").

10. Item 14: "user system being assumed to have a user buffer for receiving media data and facilities to play back the streaming media"

This term appears in claims 1, 3, 8, 9, and 15 of the '611 Patent, and in claim 2 (via dependence from claim 1) of the '839 Patent. In each case, this limitation appears in the preambles of the respective claims, reciting the intended use of the claimed invention. In this context, it is clear that this limitation is simply reciting aspects that will provide background for the body of the claim. That is, that the limitations recited in the body have, as background, the underlying assumption stated in the preamble that the user system has a user buffer for receiving media data and facilities to play back the streaming media, *e.g.*, a media player.

Whether the server software checks this assumption is immaterial. Nothing in specifications of the '611 and '839 Patents contradict this understanding. Notably, there is *no disclosure* (let alone claim limitation) that the "server software *does not check* that a user system

CORR CRONIN LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

has a user buffer for receiving media data and facilities to play back the streaming media," as FCI would read into the claim.

According to the specification, "[i]nitially, the user buffer manager 22 requests the server 12 to send media data elements to start the playback stream, such as by selecting a radio station from a list. The server 12 responds by sending data elements to the user computer 18 at higher than the playback rate, until the entire FIFO buffer 14 has been sent to the user computer." '611 Patent, 9:18-23; '839 Patent, 9:18-23. To the extent that, in one disclosed embodiment, it is the user buffer manager that sends a request for media data elements, such a request would actually serve as an implicit confirmation (and thus check) by the server software that the user system has a user buffer and facilities for playback – and from which the server can then assume that such features are present in the user system. FCI's construction thus contradicts the specification and must therefore be incorrect.

11. Item 15: "from a server assumed to be capable of sending streaming media elements at a rate more rapid than the rate at which said streaming media is played back by a user"

This term appears in claim 24 '141 Patent, and like the term in Section 10 above appears in the preamble of claim 24, reciting the intended use of the claimed invention. As such, a POSITA would recognize that this limitation is reciting aspects that set the background for steps recited in the body of the claim.

The '141 Patent specification is consistent with this understanding. Certainly, there is no disclosure that the "media player software *does not check* that the server is capable of sending streaming media elements at a rate more rapid than the rate at which said streaming media is played back by a user," as proposed by FCI. Rather, it is clear from both the claims and the specification that this is simply an assumption.

CORR CRONIN LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

Exhibit 18
to
Ex Parte Reexamination
of
U.S. Patent No.
8,327,011

HONORABLE BARBARA J. ROTHSTEIN

UNITED STATES DISTRICT COURT
WESTERN DISTRICT OF WASHINGTON
AT SEATTLE

WAG ACQUISITION, LLC, Plaintiff, v. FLYING CROCODILE, INC., d/b/a FCI, INC., et al. Defendants.	Case No. 2:19-cv-01278-BJR **PLAINTIFF'S RESPONSIVE CLAIM CONSTRUCTION BRIEF**

CORR CRONIN LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

7. Item 11: The "playback rate" limitation

FCI attempts to rewrite simple terms without any justification. The *only* citation to the patents that FCI relies upon that even references bitrate (*i.e.*, bits per unit time as FCI would have it) is in a section discussing transport layer bandwidth capabilities for uninterrupted streaming, and thus is not equivalent to (or is a subset of) the application layer "playback rate." *See* FCI OBr. at 21–22 (citing '141 patent (D.I. 261-2 at 2–14), 1:63–66)). Yet, as discussed exhaustively elsewhere, both WAG and FCI understand that the claims are directed to the application layer, not the transport layer. Every other citation FCI relies upon actually supports WAG's construction or is at least neutral. *See id.* (bulleted citations and following).

As with the other "rate" limitations, FCI begins its argument with the unsupported assertion that "the term 'rate' by itself does not provide a measure by which a factfinder can evaluate infringement." *Id.* at 21. "Normal" refers to the rate of the presentation, and to it not being sped up or in slow motion. FCI's arguments that "normal" is indefinite is without substance, since a POSITA understands that the presentation is represented by the encoding. *See Sonix Tech. v. Publications, Intern.*, 844 F.3d 1370, 1376–77 (Fed. Cir. 2017) (finding "visually negligible" not indefinite since "a skilled artisan would understand, with reasonable certainty, what it means . . . .").

FCI's allegation that "WAG's proposed construction of playing out media 'in a normal rendition' finds no support within the Asserted Patents," is disingenuous. FCI OBr. at 22. The '611 patent discloses: "[i]n conventional systems for streaming media over the Internet, media data (whether real-time or file based) is simply transmitted from the server to the user at the rate at which it will be played out (the 'playback rate'), regardless of the data rate capabilities of the connection between the server and the user." '611 patent, 5:36-42. This excerpt distinguishes between "data rate capabilities of the connection between the server and the user," that is, transport

CORR CRONIN LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

layer details as present in FCI's construction, and the playback that the user sees, that is, the application layer "playback rate," thus further highlighting FCI's faulty construction.

8. Item 12: The "sending rate" limitation

FCI's statement that "Defendant's construction amplifies how rates are measured in the data transfer context" (FCI OBr. At 23) really means that FCI's construction unduly limits the term. This quote also shows that FCI is seeking to again inject transport layer technicalities into claims directed to application layer functionality.

In justifying why the technicalities of "bits per unit time" must be injected into this term, FCI states that "[d]efining 'rate' in an objective measure of bits over time will assist the factfinder in making a rate assessment when comparing sending rates to playback rates." *Id.* at 23. But FCI never explains why a fact finder cannot just compare the sending rate to the playback rate, rather than being forced to analyze transport layer details of bitrates. Nor does FCI explain away the disclosure in the specification that "[t]he server buffer 14 'sends' data by delivering it to the transport mechanism," indicating that the sending rate is a function of something in addition to the underlying transport layer bit rate. '611 patent, 8:14–15. As for FCI's citation to arguments made in an IPR, the quote relied upon compares sending rate to playback rate, with an ancillary note that the underlying transport mechanism must be able to support transmissions faster than the playback rate; nothing about this forces the conclusion that the sending rate must be a bit rate. In short, FCI has pointed to no evidence in the intrinsic record to limit the sending rate to only a bit rate.

9. Item 13: "serial number / identifier"

Nothing in the plain language of these two terms requires them to be either consecutive or limited to numerals, as argued in WAG's opening brief. *See* D.I. 263 at 25–26. Serial items indicate a series and requiring them to be consecutive would render superfluous claim 2 of the '141 patent. Just because there is a "series" of serial numbers / identifiers does not mean that they must be consecutive. Non-consecutive numbers can represent a series.

CORR CRONIN LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

WAG did not argue in the IPRs that systems using pre-existing identifiers were "excluded from the scope of the Asserted Patents." *See* FCI OBr. at 24 (citing IPR 2015-01037, Paper 6 at 30). This argument was directed to claims 19–23 of the '141 patent, which, unlike the claims here, specifically recite a routine to "serially identify" sequential data elements, *i.e.*, the act of identifying them. D.I. 262 at 260 (heading above FCI's quote, ending in "Claims 19–23."). WAG also never argued that the "identifier" in claims 1 and 24 of the '141 patent is a "serial number" as FCI argues. *See* FCI OBr. at 24. The asserted prior art included "serial numbers" and WAG was distinguishing these from the identifiers of the claims using the terminology of the asserted art itself.

Moreover, FCI tries to conflate the fact that the media data elements themselves are "sequential" with some requirements that the identifiers themselves are consecutive. The series of the identifiers / numbers *corresponds* to a sequence. That sequence is the multimedia data itself. All multimedia data is in a time sequence. The identifiers / numbers are in a series that identifies that sequence, but the identifiers / numbers are not themselves necessarily in a sequence or consecutive, they are merely in a defined series so that they correspond to the sequence of the multimedia data elements. They could even be random. Finally, FCI's cite to the Oxford Dictionary (FCI OBr. at 25) shows that WAG's definition is correct. An item that shows a position in a series is perfectly in-line with identifiers that correspond to a sequence. This definition in no way supports the more limiting "consecutive." It suggests just the opposite – that the identifiers / numbers need <u>not</u> be consecutive.

10. Item 14: "user system being assumed to have a user buffer for receiving media data and facilities to play back the streaming media"

FCI argues, based on dictionary definitions, that this clause from the claim preambles must be a specific, and negative limitation on the steps that the overall system takes. It is not so. As

CORR CRONIN LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

argued in WAG's opening brief (D.I. 263 at 26–27), this term merely means that the system is constructed under the assumption that the user system has a buffer with the stated properties.

11. Item 15: "from a server assumed to be capable of sending streaming media elements at a rate more rapid than the rate at which said streaming media is played back by a user"

As with Item 14 above, there is no specific negative requirement in this language.

12. Item 17: "about the playback rate"

The Parties nearly agree with the construction of this term, except that FCI, again, insists upon inserting the further limitation that "'rate' cannot be met/measured using aggregates or averages," which is pure surplusage that finds no support in the claim language itself or in the specification. As discussed above in relation to Item 2, WAG's prior arguments regarding averages are not applicable. One went to "long term averages" and the other was not actually an argument about averages, but rather about an "aggregate."

The Court should thus adopt the plain meaning for this term, *e.g.*, at approximately the playback rate, which is the rate at which the media will be played out in a normal rendition. *See Modine Mfg. Co. v. U.S. Int'l Trade Comm'n*, 75 F.3d 1545, 1554 (Fed. Cir. 1996), *abrogated on other grounds by Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.*, 234 F.3d 558 (Fed. Cir. 2000) (terms such "as 'about' must be given reasonable scope . . . [and] viewed by the decisionmaker as they would be understood by persons experienced in the field of the invention").

CONCLUSION

For the forgoing reasons, the Court should construe each of the above-noted limitations to have its plain and ordinary meaning, which interpretation is consistent with the internal language of the claims themselves and the specification.

CORR CRONIN LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/014,833	08/25/2021	8327011	125737.538655	1250

7590 11/16/2021

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES, LLC
231 SOMERVILLE ROAD
BEDMINSTER, NJ 07921

EXAMINER	
WASSUM, LUKE S	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
11/16/2021	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

VENABLE LLP
1290 AVENUE OF THE AMERICAS
NEW YORK, NY 10104-3800

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. *90/014,833* .

PATENT UNDER REEXAMINATION *8327011* .

ART UNIT *3992* .

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

PTOL-465 (Rev.07-04)

Order Granting Request For Ex Parte Reexamination	**Control No.** 90/014,833	**Patent Under Reexamination** 8327011	
	Examiner Luke S Wassum	**Art Unit** 3992	**AIA (FITF) Status** No

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

The request for *ex parte* reexamination filed 08/25/2021 has been considered and a determination has been made. An identification of the claims, the references relied upon, and the rationale supporting the determination are attached.

Attachments: a)☐ PTO-892, b)☑ PTO/SB/08, c)☐ Other: ____

1. ☑ The request for *ex parte* reexamination is GRANTED.

RESPONSE TIMES ARE SET AS FOLLOWS:

For Patent Owner's Statement (Optional): TWO MONTHS from the mailing date of this communication (37 CFR 1.530 (b)). **EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c).**

For Requester's Reply (optional): TWO MONTHS from the **date of service** of any timely filed Patent Owner's Statement (37 CFR 1.535). **NO EXTENSION OF THIS TIME PERIOD IS PERMITTED.** If Patent Owner does not file a timely statement under 37 CFR 1.530(b), then no reply by requester is permitted.

/ANGELA M LIE/ Primary Examiner, Art Unit 3992		/LUKE S WASSUM/ Primary Examiner, Art Unit 3992

cc:Requester (if third party requester)

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/014,833	08/25/2021	8327011	125737.538655	1250

7590 03/04/2022

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES, LLC
231 SOMERVILLE ROAD
BEDMINSTER, NJ 07921

EXAMINER	
WASSUM, LUKE S	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
03/04/2022	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

VENABLE LLP
1290 AVENUE OF THE AMERICAS
NEW YORK, NY 10104-3800

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. *90/014,833* .

PATENT UNDER REEXAMINATION *8327011* .

ART UNIT *3992* .

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

PTOL-465 (Rev.07-04)

Priority

The ‘011 patent is a continuation of U.S. Patent Application 12/800,152, filed 10 May 2010, now U.S. Patent 8,122,141, which is a continuation of U.S. Patent Application 10/893,814, filed 19 July 2004, now U.S. Patent 7,716,358, which is a continuation-in-part of U.S. Patent Application 09/819,337, filed 28 March 2001, now U.S. Patent 6,766,376.

The ‘011 patent also claims priority under 35 U.S.C. § 119(e) to U.S. Provisional Application 60/231,997, filed 12 September 2000.

Therefore, the earliest claimed priority date of the ‘011 application is 12 September 2000, *to the extent that the claimed subject matter is fully supported*.

Claim Interpretation

The Office notes that the ‘011 patent contains a specific reference to several earlier filed applications to which priority is claimed under 35 U.S.C. § 120. Therefore, since the patent term is twenty years from the filing date of the earliest such application (in this case, 28 March 2001), the ‘011 patent expired as of 28 March 2021.

In a reexamination proceeding involving claims of an expired patent, claim construction pursuant to the principle set forth by the court in *Phillips v. AWH Corp.*, 415 F.3d 1303, 1316, 75 USPQ2d 1321, 1329 (Fed. Cir. 2005) (words of a claim "are generally given their ordinary and customary meaning" as understood by a person of ordinary

skill in the art in question at the time of the invention) should be applied since the expired claim are not subject to amendment. See *Ex parte Papst-Motoren*, 1 USPQ2d 1655 (Bd. Pat. App. & Inter. 1986).

Upon review of the original specification and prosecution history, the examiner has found no instance of lexicographic definitions, either express or implied, that are inconsistent with the ordinary and customary meaning of the respective terms. Therefore, for the purposes of claim interpretation, the examiner concludes that there are no claim terms for which Patent Owner is acting as their own lexicographer. See MPEP § 2111.01(IV).

The Claimed Invention

The invention provides a system for distributing via the Internet streaming media composed of a plurality of time-sequenced data elements. The system has a server connected to the Internet for transmitting the data elements. Associated with the server are a buffer manager and a FIFO buffer for storing at least one of the data elements for transmission. The buffer manager comprises means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the FIFO buffer with a predetermined number of data elements; maintaining a pointer into the

playback rate in **Shteyn** to enhance **Shteyn's** functionality, including real-time playback. **Shteyn** at Abstract, 1:62-66. A POSITA would have been further motivated to implement such function considering **Shteyn's** existing server structure and buffering ability to store the data sent faster than the playback rate. **Shteyn** at Abstract, 1:62-66, 3:14-18, 3:31-36, 4:20-31. Moreover, **Shteyn** has an existing system, just like **Glaser**, to receive the appropriate quality data. Compare **Shteyn**, at 3:44-53, 3:57-61, 4:20-26, Fig. 1, with **Glaser**, at 21:57-63, 22:41-59.

Accordingly, a POSITA would have been motivated to combine **Shteyn** and **Glaser**.

<u>Ground 3a: Hill</u>

Claims 1 and 4 are anticipated under 35 U.S.C. § 102(e) in view of **Hill**.

Regarding claim 1, **Hill** teaches **a media player** (see disclosure that the invention relates to audio/video playback devices, col. 1, lines 7-11 et seq.) for receiving **an audio or video program** as claimed, **the program comprising media data elements** (see disclosure that movie data is composed of synchronized audio and video data broken into fundamental units called frames, col. 3, lines 37-39 et seq.; see also disclosure of

clips, which are a sequence of frames, col. 3, lines 46-47 et seq.), from **a media source over an Internet protocol network** (see disclosure that clips may be stored in different locations within the digital network environment, that movie data contains a number of clips stored in separate sources, and that a source may be a site on the Internet, col. 3, lines 48-56 et seq.), and **playing the program for a user of the media player** (see disclosure that the user utilizes the workstation to play the movie data, col. 3, lines 28-29 et seq.), wherein **each of the media data elements is associated with a serial number** (see disclosure that frames are requested with respect to a frame number (i.e., serial number), col. 5, line 59 through col. 6, line 16 and col. 7, lines 39-50 et seq.; see also disclosure of the use of frame numbers (i.e., serial numbers), for identifying specific frames, Figs. 6B and 7:

REEXAMINATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Harold Edward Price	Docket No.: 125737.538655
U.S. Patent No.:	8,327,011	Group Art Unit: 3992
Reexamination:	90/014,833	Confirmation No.: 1250
Filed:	August 25, 2021	Examiner: WASSUM, Luke S.
For:	Streaming Media Delivery System	

RESPONSE TO OFFICE ACTION UNDER 37 C.F.R. § 1.530

Mail Stop *Ex Parte* Reexam
ATTN: Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Examiner Wassum:

The Patent Owner responds to the non-final Office action mailed March 4, 2022 ("NFOA") for the above-captioned *Ex Parte* Reexamination proceeding, as follows:

Argument commences on page 2.

ARGUMENT

Claims 1 and 4 of U.S. Patent No. 8,327,011 (the "'011 patent") are under reexamination and stand rejected under the NFOA. Patent Owner addresses herein the issues raised in the NFOA.

Patent Owner observes that NFOA follows on a lengthy third-party request for *ex parte* reexamination of the claims (the "Request"), and tracks many of the Request's characterizations of the prior art as well as its interpretation of the claim terms. Patent Owner respectfully submits that the Request overstated many teachings of the cited references and reads a number of the claim limitations beyond the literal claim language as interpreted in light of the specification, to unduly ease the task in challenging the claims.

Before addressing the individual grounds of the NFOA, Patent Owner addresses some preliminary points common to the issues addressed herein.

Claim Construction—*Phillips* and not BRI. As the NFOA notes, the '011 patent has expired and thus the proper claim construction standard is under *Phillips v. AWH* rather than the "broadest reasonable interpretation" ("BRI"). MPEP 2258(I)(G). Interpretations based on BRI must therefore be avoided because they lead to overly broad constructions and potentially to rejections that are not properly supported.

Federal Circuit Decision. The Request as well as the NFOA cites the PTAB decision on remand in IPR2016-0123 (previously submitted with the Request as Ex. 12), concerning different claims of a different patent, U.S. Patent No. 8,122,141 (Ex. 1 to the request). Significantly, however, the Request omitted to submit the decision of the Federal Circuit on which the remand was based, in which the Federal Circuit had reversed the earlier decision of the PTAB. The Federal Circuit's decision, reported at 781 Fed. Appx. 1007 (Fed. Cir. 2019) is submitted herewith as Patent Owner's Ex. 101.

Patent Owner respectfully requests that the Examiner review and consider the Federal Circuit decision (Ex. 101 herewith), in addressing the scope of the claims herein. The Federal Circuit decision contains important guidance on claim interpretation that is significant to the present reexamination, particularly its ruling that the relevant "rate" of data transfer under the claims at issue in that case was "the rate at which **each** requested media data element is transmitted from the server to the user computer." Ex. 101 at 10 (781 Fed. Appx. 1007, 1011 (Fed. Cir. 2019)). That ruling, plus the difference in wording between the claims at issue in that

case and the claims here, are significant factors that must be taken into account in interpreting the present claims and considering them in view of the prior art.

Patents and Printed Publications. Reexamination is of course limited to patents and printed publications. 37 CFR 1.552; MPEP 2258. Patent Owner brings up this point because the third-party Request repeatedly makes assertions based on deposition testimony and the like, which are not patents or printed publications. Review procedures, including both IPRs and *ex parte* reexaminations, are both limited to patents and printed publications as bases for finding invalidity. It is important, in complying with this requirement, not to allow testimonial assessments, such as by depositions and declarations, to become "a replacement for documentary evidence for core factual findings in a determination of patentability." *K/S HIMPP v. Hear-Wear Technologies*, LLC, 751 F.3d 1362, 1365-66 (Fed. Cir. 2014). Such submissions are useful only in the narrow circumstances that enable an "instant and unquestionable demonstration as being well-known." *Id.* at 1366. Where limitations are not disclosed in the cited patents or printed publications, it would be improper to rely on expert declarations or deposition testimony as a substitute, which the Federal Circuit described, with reference to proceedings including reexamination in the CRU, as "a slippery slope which would permit the examining process to deviate from the well-established and time-honored requirement that rejections be supported by evidence." *Id.*

ABBREVIATIONS FOR CERTAIN MISSING LIMITATIONS

To avoid excessive verbiage, yet still be precise, the following shorthand abbreviations and reference labels are provided to stand for the full text of various claim limitations asserted herein to be missing from the disclosures of the cited references ("**MLs**"). References herein to specific MLs are intended to reference the entire text of the corresponding ML, as set forth in the following table. Instead of repeating the full claim language at length over and over again in the following discussion, the abbreviations set forth below will be used.

Ref.	Abbreviation	Full Text of Limitation
ML1-1	"**maintain a record of the serial number of the last media data element**"	[Claim 1] instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to **maintain a record of the serial number of**

Office Action in Ex Parte Reexamination	Control No. 90/014,833	Patent Under Reexamination 8327011	
	Examiner Luke S Wassum	Art Unit 3992	AIA (FITF) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

a. ☑ Responsive to the communication(s) filed on 06 June 2022.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on ______.

b. ☑ This action is made FINAL.

c. ☐ A statement under 37 CFR 1.530 has not been received from the patent owner.

A shortened statutory period for response to this action is set to expire 2 month(s) from the mailing date of this letter.
Failure to respond within the period for response will result in termination of the proceeding and issuance of an *ex parte* reexamination certificate in accordance with this action. 37 CFR 1.550(d). **EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c).**
If the period for response specified above is less than thirty (30) days, a response within the statutory minimum of thirty (30) days will be considered timely.

Part I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

1. ☐ Notice of References Cited by Examiner, PTO-892.
2. ☐ Information Disclosure Statement, PTO/SB/08.
3. ☐ Interview Summary, PTO-474.
4. ☐ ______.

Part II SUMMARY OF ACTION

1a. ☑ Claims 1 and 4 are subject to reexamination.
1b. ☑ Claims 2-3 are not subject to reexamination.
2. ☐ Claims ______ have been canceled in the present reexamination proceeding.
3. ☐ Claims ______ are patentable and/or confirmed.
4. ☑ Claims 1 and 4 are rejected.
5. ☐ Claims ______ are objected to.
6. ☑ The drawings, filed on 25 August 2021 are acceptable.
7. ☐ The proposed drawing correction, filed on ______ has been (7a) ☐ approved (7b) ☐ disapproved.
8. ☐ Acknowledgment is made of the priority claim under 35 U.S.C. 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the certified copies have
   1 ☐ been received.
   2 ☐ not been received.
   3 ☐ been filed in Application No. ______.
   4 ☐ been filed in reexamination Control No. ______.
   5 ☐ been received by the International Bureau in PCT application No. ______.
   * See the attached detailed Office action for a list of the certified copies not received.
9. ☐ Since the proceeding appears to be in condition for issuance of an *ex parte* reexamination certificate except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte* Quayle, 1935 C.D. 11, 453 O.G. 213.
10. ☐ Other: ______

cc: Requester (if third party requester)

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/014,833	08/25/2021	8327011	125737.538655	1250

7590 07/27/2022

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES, LLC
231 SOMERVILLE ROAD
BEDMINSTER, NJ 07921

EXAMINER	
WASSUM, LUKE S	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
07/27/2022	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

VENABLE LLP
1290 AVENUE OF THE AMERICAS
NEW YORK, NY 10104-3800

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. *90/014,833* .

PATENT UNDER REEXAMINATION *8327011* .

ART UNIT *3992* .

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

PTOL-465 (Rev.07-04)

Ex Parte Reexamination

A request for *Ex Parte* Reexamination of claims 1 and 4 of United States Patent Number 8,327,011 ("the '011 patent") was filed 25 August 2021 ("the Request"). The '011 patent issued to **Harold Edward Price** on 4 December 2021, from application number 13/374,942 ("the '942 application"), filed 24 January 2012.

The Office mailed a Decision granting reexamination of claims 1 and 4 on 16 November 2021.

Claims 1 and 4 are subject to reexamination.

Related Proceedings

After examiner's independent review of the '011 patent and its prosecution history, the examiner notes that there is current ongoing litigation (detailed on the Reexamination sheet, document code RXFILJKT). The examiner has failed to locate any previous reexaminations (*ex parte* or *inter partes*). However, the '011 patent was the subject of three prior *Inter Partes* Review proceedings: IPR 2015-01033, IPR 2016-01161, and IPR 2016-01655. The PTAB denied institution in all 3 IPR proceedings.

Priority

The '011 patent is a continuation of U.S. Patent Application 12/800,152, filed 10 May 2010, now U.S. Patent 8,122,141, which is a continuation of U.S. Patent Application 10/893,814, filed 19 July 2004, now U.S. Patent 7,716,358, which is a continuation-in-part of U.S. Patent Application 09/819,337, filed 28 March 2001, now U.S. Patent 6,766,376.

The '011 patent also claims priority under 35 U.S.C. § 119(e) to U.S. Provisional Application 60/231,997, filed 12 September 2000.

Therefore, the earliest claimed priority date of the '011 application is 12 September 2000, *to the extent that the claimed subject matter is fully supported*.

Claim Interpretation

The Office notes that the '011 patent contains a specific reference to several earlier filed applications to which priority is claimed under 35 U.S.C. § 120. Therefore, since the patent term is twenty years from the filing date of the earliest such application (in this case, 28 March 2001), the '011 patent expired as of 28 March 2021.

In a reexamination proceeding involving claims of an expired patent, claim construction pursuant to the principle set forth by the court in *Phillips v. AWH Corp.*, 415 F.3d 1303, 1316, 75 USPQ2d 1321, 1329 (Fed. Cir. 2005) (words of a claim "are generally given their ordinary and customary meaning" as understood by a person of ordinary

skill in the art in question at the time of the invention) should be applied since the expired claim are not subject to amendment. See *Ex parte Papst-Motoren*, 1 USPQ2d 1655 (Bd. Pat. App. & Inter. 1986).

Upon review of the original specification and prosecution history, the examiner has found no instance of lexicographic definitions, either express or implied, that are inconsistent with the ordinary and customary meaning of the respective terms. Therefore, for the purposes of claim interpretation, the examiner concludes that there are no claim terms for which Patent Owner is acting as their own lexicographer. See MPEP § 2111.01(IV).

The Claimed Invention

The invention provides a system for distributing via the Internet streaming media composed of a plurality of time-sequenced data elements. The system has a server connected to the Internet for transmitting the data elements. Associated with the server are a buffer manager and a FIFO buffer for storing at least one of the data elements for transmission. The buffer manager comprises means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the FIFO buffer with a predetermined number of data elements; maintaining a pointer into the

buffer for each user computer indicating the last media data element that has been sent to that user, thus indicating the next element or elements to be sent; and, once the FIFO buffer is full, deleting the oldest data elements in the buffer as new data elements are received, said means arranged to maintain the pre-determined number of data elements in the FIFO buffer. At least one user computer is connected to the server via the Internet or other data communications medium.

The server buffer manager, or the media source, provides for sequentially numbering the media data elements. The server buffer manager does not maintain a pointer into the server buffer for each user. Instead, the media player buffer manager in the user computer maintains a record of the serial number of the last data element that has been received. Via the use of standard data communications protocol techniques such as TCP, the user computer transmits a request to the server to send one or more data elements, specifying the serial numbers of the data elements. The server responds by sending the requested data elements, and depends upon the reliable transmission protocol to assure delivery. The user computer then continues with additional data requests for the duration of playing the audio/video material. In this manner, the user computer, not the server, maintains the record of the highest data element number stored in the user computer buffer. The media data will be transmitted to the user

computer as fast as the data connection between the user computer and the server will allow.

Prior Art Cited in the Request

The following reference was cited by the third party Requester in the Request for *Ex Parte* Reexamination:

U.S. Patent 7,529,806 to Shteyn ("**Shteyn**")

U.S. Patent 6,005,600 to Hill ("**Hill**")

U.S. Patent 6,175,862 to Feig et al. ("**Feig**")

U.S. Patent 6,389,473 to Carmel et al. ("**Carmel**")

U.S. Patent 6,792,468 to Bloch et al. ("**Bloch**")

International Application WO 1997044942 to Kliger et al. ("**Kliger**")

U.S. Patent 5,987,510 to Imai et al. ("**Imai**")

U.S. Patent 5,793,980 to Glaser et al. ("**Glaser**")

Patent Owner's Response

A response to the non-final Office action ("Response") was filed by Patent Owner on 6 June 2022.

The response included remarks and a supporting exhibit 101, the Federal Circuit's decision on appeal of the PTAB's Final Written Decision in IPR2016-01238.

Response to Arguments

Patent Owner has presented a number of arguments with respect to the rejected claims. These arguments are addressed in turn below.

Patent Owner presented general arguments regarding the contents of the Request having overstated the teachings of the cited references, the applied claim construction, in particular with respect to the interpretation of the claim term "rate", and that the rejections improperly rely upon expert declarations or depositions instead of relying upon those limitations disclosed in the prior art references.

These arguments are taken under consideration.

<u>Ground 1: **Shteyn**</u>

With respect to the claim rejections of Ground 1, Patent Owner argues that the claim limitation "to transmit to the media source a request to send one or more media data elements, each identified by a serial number", requires that the serial number is

incorporated in the request. Patent Owner further argues that **Shteyn** fails to disclose this limitation, but only discloses that the client requests that the server download the next segment, and does not specify a particular serial number in the request.

The Office respectfully disagrees.

Shteyn discloses that media segments are available on the server in a variety of formats ("part1" vs. "part1 alt"). There is an explicit disclosure that during operation, the client application could select a next segment in a different format for the same content to adapt to changing circumstances, e.g., lower bandwidth due to network congestion (col. 4, lines 20-23).

If **Shteyn's** client application could only make a request to download the 'next' segment, it could not have the disclosed capability to download an alternate version of the segment. In order to do so, the request would have to include the claimed 'serial number' (interpreted as argued by Patent Owner in previous litigation, i.e., that the claimed serial number is not limited to consecutive numbers or limited to numerals), which indicates which specific segment is being requested by the client application.

Were **Shteyn's** client application limited to requesting only the 'next' segment, then once the first segment has been requested in a specific format, the client application would be limited to requesting the next segment in the same format.

Patent Owner also argues that **Shteyn** fails to disclose the claimed limitation of maintaining a record of the serial number of the last media data element that has been received.

The Office respectfully disagrees.

Shteyn discloses a process flowchart in Fig. 1, whereby media data segments are downloaded and rendered in turn. In particular, step 110 discloses the step of downloading the next file segment and buffering it while rendering the preceding segment.

In order for the client application to know which segment is to be rendered, and which segment is the 'next' to be downloaded, it would have been necessary for the application to maintain a record of which segment was the last one to have been downloaded. Since the XML file stored on the client maintains information about the sequence of media segments, and other information including a network location from which each segment can be downloaded, **Shteyn** must necessarily maintain a record of

the serial number of the last downloaded segment, since this would be necessary in order for the client application to know which segment is the 'next' segment to be downloaded. Were this not to be the case, the system disclosed by **Shteyn** would be unable to function properly.

Patent Owner then argues that with respect to claim 4, **Shteyn** fails to disclose the claimed reception of media segments at a rate that is more rapid than the playback rate.

The Office respectfully disagrees.

Shteyn discloses a mechanism whereby a given segment can be available in different formats, such that if, for example, network congestion is limiting the available bandwidth, an alternate version of subsequent segments that require less bandwidth could be requested (col. 1, lines 28-31, col. 4, lines 20-23, as well as Fig. 2, illustrating different versions of a segment having different bandwidth requirements).

An ordinary artisan at the time of the invention would have understood this disclosure as providing functionality that would allow the media content download speed to exceed the rendering speed. Were this not to be the case, the system disclosed by **Shteyn** would not work correctly, since there could be gaps in the rendering of the

media content when rendering of the previous segment has completed while downloading of the next segment continues.

Patent Owner argues that this disclosure of **Shteyn** teaches the reduction of program quality, and not the reception of segments more rapidly than the playback rate. However, regardless of whether this could be seen as "a fallback approach", an ordinary artisan would have understood that the end result is to provide a mechanism that ensures the capability of receiving media segments at a rate that exceeds the playback rate, in order to ensure that the media can be rendered in a continuous fashion without gaps.

The Office also notes that claim 4 is directed to software operating on the client media player, and specifically to the rate at which media data elements are received. However, one could not write software that would guarantee that data is received at any given rate, since that rate is fully dependent upon the rate at which the data is transmitted, and further upon network conditions such as bandwidth and congestion.

At best, claim 4 requires that the software on the client media player is capable of receiving data at a rate exceeding the playback rate, in those cases where the transmission rate is sufficiently high to permit such reception speed. It would not be possible to write software that receives data at a rate greater than the rate at which it is

transmitted, and <u>the claims are silent with respect to the *transmission* rate of the media data elements</u>.

For these reasons, Patent Owner's reliance upon the Federal Circuit's decision in IPR2016-01238 for the construction of the 'rate' limitation is inapplicable in this case, since the claims of the '141 patent are directed to the rate of *transmission*, which is fully under the control of software on the server, while the instant claims are directed to the rate of *reception*, which is not controlled by the software running on the media player client, but is instead dependent upon the transmission rate and network conditions.

The rejection of claims 1 and 4 is <u>maintained</u>.

<u>Ground 2a: **Shteyn** and **Carmel**</u>

With respect to the claim rejection of Ground 2a, Patent Owner argues that **Carmel**, like **Shteyn**, discloses a system where perhaps an overall reception rate is greater than the playback rate, but it fails to disclose a system whereby <u>each</u> media data element is received at a rate faster than the playback rate.

For the reasons discussed above with respect to Ground 1, the Office respectfully disagrees.

As noted above, because the instant claims are directed to software on the client media player, and because the rate of reception of media data elements is completely dependent upon the transmission rate and network conditions, claim 4 cannot be interpreted as requiring the reception rate of each media data element to exceed the playback rate. At best, the software is required to be capable of receiving the media data elements at a rate higher than the playback rate.

Carmel's specification at col. 11, lines 9-22 with respect to the data rate (i.e., that the data rate can be increased by changing the data compression level) by itself discloses that the software on the client device is capable of receiving the media data at a rate that exceeds the playback rate, as required by the language of claim 4.

The rejection of claim 4 is maintained.

Ground 2b: **Shteyn** and **Glaser**

With respect to the claim rejection of Ground 2b, Patent Owner argues that **Glaser**, like **Shteyn**, discloses a system where perhaps an overall reception rate is greater than the playback rate, but it fails to disclose a system whereby each media data element is received at a rate faster than the playback rate.

For the reasons discussed above with respect to Ground 1, the Office respectfully disagrees.

As noted above, because the instant claims are directed to software on the client media player, and because the rate of reception of media data elements is completely dependent upon the transmission rate and network conditions, claim 4 cannot be interpreted as requiring the reception rate of each media data element to exceed the playback rate. At best, the software is required to be capable of receiving the media data elements at a rate higher than the playback rate.

Glaser discloses at col. 22, lines 57-59, that normal quality data blocks are received at a rate greater than real time, and so the buffer eventually refills and approaches maximum capacity. This disclosure, by itself, demonstrates that **Glaser's** software on the client device is capable of receiving media data at a rate greater than the playback rate (i.e., greater than real time), as required by claim 4.

Patent Owner also argues that **Glaser's** architecture does not even address a "pull mechanism" as is claimed. However, the Office notes that **Glaser** is not relied upon to teach mechanisms for initiating/requesting transmission of media data; **Shteyn** teaches these features.

Glaser is relied upon only for its disclosure of mechanisms for ensuring that the media data is received at a rate that exceeds the playback rate, which is taught with respect to the transmission of normal quality data blocks (col. 22, lines 57-59).

The rejection of claim 4 is <u>maintained</u>.

<u>Ground 3a: **Hill**</u>

With respect to the claim rejection of Ground 3a, Patent Owner argues that **Hill** fails to disclose the claimed request by serial number to the media server.

The Office respectfully disagrees.

Hill discloses that if new frames are required, the buffer manager uses the cliplist manager to retrieve those frames from the appropriate source (col. 5, lines 24-28). Also

disclosed is the fact that when a request for new frames is issued to the cliplist manager, it converts the global frame number to a clip number and a local frame number, and then passes the request to the appropriate source manager which retrieve the requested frame data from the corresponding source (col. 6, lines 7-16).

Notably, **Hill** discloses that while performing the retrieval, the source access the *requested* frame (col. 5, lines 29-30). Since a given source stores any number of frames, a request for a frame would necessarily include information to identify the requested frame (such as the local frame number that has been decoded by the cliplist manager from the global frame number in the request from the buffer manager).

See also exemplary disclosure that in the case where global frame 106 is required, cliplist manager maps global frame 106 to clip 2, local frame 17, *then requests* <u>*this frame*</u> from the second source manager (col. 12, lines 31-34).

Patent Owner also argues that **Hill** fails to disclose the claimed feature of maintaining a record of the serial number of the last media data element that has been received, but instead only maintains a record of the last media data element *requested*.

The Office respectfully disagrees.

Hill discloses that the system keeps track of not only the last media data element requested, but also the last one received.

For instance, at col. 12, lines 27-40, **Hill** discloses that the buffer manager checks to see if global frames 96 through 106 are currently being stored, *or whether they have been requested*. This means that the system differentiates between requested frames and *received* frames, by maintaining records of not only which frame has been last requested, but which has been last received (and stored in the buffer). When the frame is received, the buffer manager inserts the received frame into the buffer as global frame 106 (and updates its record to reflect the insertion of global frame 106 into the buffer.)

Patent Owner also argues that **Hill** discloses a system where perhaps an overall reception rate is greater than the playback rate, but it fails to disclose a system whereby <u>each</u> media data element is received at a rate faster than the playback rate.

For the reasons discussed above with respect to Ground 1, the Office respectfully disagrees.

As noted above, because the instant claims are directed to software on the client media player, and because the rate of reception of media data elements is completely dependent upon the transmission rate and network conditions, claim 4 cannot be

interpreted as requiring the reception rate of each media data element to exceed the playback rate. At best, the software is required to be <u>capable of</u> receiving the media data elements at a rate higher than the playback rate.

Hill's specification at col. 7, lines 11-24, that the system can download multiple frames, such that if the playback rate is 24 frames/second, the download rate can be up to 48 frames/second, by itself discloses that the system is capable of receiving the frame data at a rate that exceeds the playback rate, as required by the language of claim 4.

The rejection of claims 1 and 4 is <u>maintained</u>.

<u>Ground 3b: **Hill** and **Carmel**</u>

With respect to the claim rejection of Ground 3b, Patent Owner argues that **Carmel**, like **Hill**, discloses a system where perhaps an overall reception rate is greater than the playback rate, but it fails to disclose a system whereby <u>each</u> media data element is received at a rate faster than the playback rate.

For the reasons discussed above with respect to Ground 1, the Office respectfully disagrees.

As noted above, because the instant claims are directed to software on the client media player, and because the rate of reception of media data elements is completely dependent upon the transmission rate and network conditions, claim 4 cannot be interpreted as requiring the reception rate of each media data element to exceed the playback rate. At best, the software is required to be capable of receiving the media data elements at a rate higher than the playback rate.

Carmel's specification at col. 11, lines 9-22 with respect to the data rate (i.e., that the data rate can be increased by changing the data compression level) by itself discloses that the software on the client device is capable of receiving the media data at a rate that exceeds the playback rate, as required by the language of claim 4.

The rejection of claim 4 is maintained.

Ground 3c: **Hill** and **Glaser**

With respect to the claim rejection of Ground 3c, Patent Owner argues that **Glaser**, like **Hill**, discloses a system where perhaps an overall reception rate is greater than the playback rate, but it fails to disclose a system whereby each media data element is received at a rate faster than the playback rate.

For the reasons discussed above with respect to Ground 1, the Office respectfully disagrees.

As noted above, because the instant claims are directed to software on the client media player, and because the rate of reception of media data elements is completely dependent upon the transmission rate and network conditions, claim 4 cannot be interpreted as requiring the reception rate of each media data element to exceed the playback rate. At best, the software is required to be capable of receiving the media data elements at a rate higher than the playback rate.

Glaser discloses at col. 22, lines 57-59, that normal quality data blocks are received at a rate greater than real time, and so the buffer eventually refills and approaches maximum capacity. This disclosure, by itself, demonstrates that **Glaser's** software on the client device is capable of receiving media data at a rate greater than the playback rate (i.e., greater than real time), as required by claim 4.

The rejection of claim 4 is maintained.

<u>Ground 4: **Shteyn** and **Hill**</u>

With respect to the claim rejection of Ground 4, Patent Owner reiterates arguments presented previously with respect to the alleged shortcomings of the **Shteyn** and **Hill** references individually.

These arguments have been addressed above, with respect to Grounds 1 and 3a respectively.

The rejection of claims 1 and 4 are <u>maintained</u>.

<u>Ground 5a: **Feig**</u>

With respect to the claim rejection of Ground 5a, Patent Owner argues that **Feig** fails to disclose the claimed request from the media source a predetermined number of media data elements.

The Office finds this argument persuasive.

Feig discloses that media segments (analogous to the claimed media data elements) are made up of a series of URLs, such as URL(1), URL(2), etc. In order to download media segments, a "Fetch segment" module (or subsequently, a "Fetch next

segment and display present segment" module) is executed to fetch the next segment by making requests for URL(1), URL(2), and so on, until URL(A1) is received.

This being the case, there is no request made to retrieve media data elements from the media source, but only sequential requests for each individual component URL of the respective media data elements.

The rejection of claims 1 and 4 are withdrawn.

Ground 5b: **Feig** and **Carmel**

Ground 5c: **Feig** and **Glaser**

The rejections of claim 4 are withdrawn, for the same reasons discussed above with respect to the rejection of Ground 5a.

Ground 6a: **Carmel**

With respect to the claim rejection of Ground 6a, Patent Owner argues that **Carmel** fails to disclose the claimed repeated transmission of requests to the media source for sequential media data elements.

The Office respectfully disagrees.

Carmel discloses a mechanism whereby a client system connects to a server by opening one or more HTTP links. The client first reads an index file and displays a graphic to the user so they can decide the slice at which the data stream will begin downloading. Responsive to the user selection, the client selects an appropriate starting slice and begins to download the files (col. 10, lines 35-54).

Carmel also discloses the monitoring by the client of the rate of data transfer, and adjusts the quality level of media data transmitted from the server accordingly, such as by requesting media data at a lower quality level if the transfer rate is low, or requesting a higher level of quality to take advantage of available bandwidth when the transfer rate exceeds that needed to receive the successive slices on time, col. 11, lines 9-22. Adjusting the quality level would allow the system to speed or slow the rate at which slices are received, so as to ensure that the amount of media data stored in the buffer is maintained at a desired level. These requests to adjust the quality level of media data transmitted are analogous to the claimed repeated transmission of requests to the media source for sequential media data elements to maintain a predetermined number of media data elements in the player buffer.

Patent Owner also argues that **Carmel** fails to disclose the claimed receiving the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player. This argument has been addressed above with the respect to the rejection of claim 4 under Grounds 2a and 3b.

The rejection of claims 1 and 4 is maintained.

Ground 6b: **Carmel** and **Hill**

With respect to the claim rejection of Ground 6b, Patent Owner argues that **Hill** fails to disclose the claimed reception of media data elements at a rate more rapid than the playback rate.

The Office respectfully disagrees.

This argument has been previously addressed with respect to the claim rejections of Ground 3a.

The rejection of claims 1 and 4 is maintained.

Ground 7a: **Bloch**

With respect to the claim rejection of Ground 7a, Patent Owner argues that **Bloch** fails to disclose the claimed maintenance of a record of the serial number of the last media data element received.

The Office respectfully disagrees.

Bloch discloses a mechanism whereby when a client system receives a request from a user to play a media clip, the system generates frame accurate requests (e.g., Get Frame 1 of clip A"), and sends the requests to the media server. In particular, **Bloch** discloses that "each frame accurate request is associated with a single frame of movie data. Thus, in order to play a clip containing M frames, playback engine 218 generates M requests and transmits them to server 222."

Patent Owner argues that these requests are "batched", and therefore **Bloch** fails to disclose the claimed maintenance of a record of the serial number of the last media data element received, because the '011 patent uses this information to determine which media data element to request next.

The Office notes that the use of the stored serial number of the last media data element received to generate the next request is not a feature that appears in the claim language. Only the tracking of the last element received is claimed.

Bloch discloses this feature, by virtue of its disclosure that the system tracks the frames as they are received to ensure that the frames for the next clip (Clip B) are not requested until all of the frames of Clip A have been received (col. 9, lines 38-46.)

Patent Owner also argues that **Bloch** fails to disclose the claimed repeating transmission of the requests so as to maintain a pre-determined number of media data elements in the player buffer.

The Office finds this argument persuasive.

Bloch discloses the transmission of frame requests, and the resulting reception of frames from the server. Specifically, in step 610, Fig. 6, **Bloch** discloses that the client workstation receives the requested frames and renders them. There is no disclosure of maintaining a specific number of media data elements in the buffer.

Bloch discloses at col. 9, lines 15-22 that one clip could be made up of 1000 frames, while another could be made up of 500 frames. Even were one to interpret the disclosure as meaning that the system does not render the frames of clip A until all such

frames have been received, there is no disclosure of the maintenance of a *pre-determined* number of media data elements in the buffer, since a given clip could be made up of any number of frames.

The rejection of claims 1 and 4 are withdrawn.

Ground 7b: **Bloch** and **Carmel**

Ground 7c: **Bloch** and **Glaser**

The rejections of claim 4 are withdrawn, for the same reasons discussed above with respect to the rejection of Ground 7a.

Ground 8: **Kliger**

With respect to the claim rejection of Ground 8, Patent Owner argues that **Kliger** fails to disclose the claimed repeating transmission of the requests so as to maintain a pre-determined number of media data elements in the player buffer.

The Office finds this argument persuasive.

Kliger discloses a mechanism whereby when a client system receives an object stream request, the request includes a list, or linear sequence of objects identified by global object identifier numbers (page 7, lines 12-19.) However, while claim 1 calls for the client to transmit to the media source repeated requests for sequential media data elements, **Kliger** discloses the transmission of a single request in the form of a list of desired objects (page 4, lines 17-22.) Since the request for the desired objects is sent as a single request, there is no disclosure of using transmission of repeated requests for sequential media data elements to maintain a pre-determined number of media data objects in the buffer.

The rejection of claims 1 and 4 is withdrawn.

Ground 9a: **Imai**

With respect to the claim rejection of Ground 9a, Patent Owner argues that **Imai** fails to disclose the claimed repeating transmission of the requests so as to maintain a pre-determined number of media data elements in the player buffer.

The Office respectfully disagrees.

Imai discloses a mechanism whereby a number of file transfer requests are made from a client to a server. The files to be requested are maintained in a list, and the file request for each file is issued sequentially (col. 21, lines 47-53.) As can be seen in the flowchart of Fig. 24, the multiple file request unit cycles through the list, issuing requests one after another until all files have been requested. Since the request for the next file (steps S18-S20) does not take place until after the previous file has been received and stored in the cache (step S21), the system always maintains a record of the last received file.

Patent Owner also argues that **Imai** fails to disclose the claimed repeated transmission of media data element requests so as to maintain a predetermined number of media data elements in the player buffer.

The Office respectfully disagrees.

At col. 23, lines 8-27, **Imai** discloses a mechanism whereby the repeated media requests are issued until the amount of data in the cache (or alternately, a specific percentage of free space is reached), at which time requests for additional media are paused (this is reflected in the decision at step S17 of Fig. 24.) By this process, **Imai**

discloses issuing repeated media requests to the server so as to maintain a specific number of media data elements in the buffer, as claimed.

The rejection of claim 1 is maintained.

Ground 9b: **Imai** and **Carmel**

With respect to the claim rejection of Ground 9b, Patent Owner argues that **Carmel** discloses a system where perhaps an overall reception rate is greater than the playback rate, but it fails to disclose a system whereby each media data element is received at a rate faster than the playback rate.

For the reasons discussed above with respect to Ground 1, the Office respectfully disagrees.

As noted above, because the instant claims are directed to software on the client media player, and because the rate of reception of media data elements is completely dependent upon the transmission rate and network conditions, claim 4 cannot be interpreted as requiring the reception rate of each media data element to exceed the

playback rate. At best, the software is required to be capable of receiving the media data elements at a rate higher than the playback rate.

Carmel's specification at col. 11, lines 9-22 with respect to the data rate (i.e., that the data rate can be increased by changing the data compression level) by itself discloses that the software on the client device is capable of receiving the media data at a rate that exceeds the playback rate, as required by the language of claim 4.

The rejection of claim 4 is maintained.

Ground 9c: **Imai** and **Glaser**

With respect to the claim rejection of Ground 9c, Patent Owner argues that **Glaser** discloses a system where perhaps an overall reception rate is greater than the playback rate, but it fails to disclose a system whereby each media data element is received at a rate faster than the playback rate.

For the reasons discussed above with respect to Ground 1, the Office respectfully disagrees.

As noted above, because the instant claims are directed to software on the client media player, and because the rate of reception of media data elements is completely dependent upon the transmission rate and network conditions, claim 4 cannot be interpreted as requiring the reception rate of each media data element to exceed the playback rate. At best, the software is required to be <u>capable of</u> receiving the media data elements at a rate higher than the playback rate.

Glaser discloses at col. 22, lines 57-59, that normal quality data blocks are received at a rate greater than real time, and so the buffer eventually refills and approaches maximum capacity. This disclosure, by itself, demonstrates that **Glaser's** software on the client device is capable of receiving media data at a rate greater than the playback rate (i.e., greater than real time), as required by claim 4.

The rejection of claim 4 is <u>maintained</u>.

Summary of the Grounds of Rejection

<u>Ground 1</u>: Claims 1 and 4 are anticipated under 35 U.S.C. § 102(e) in view of **Shteyn**.

Ground 2a: Claim 4 is obvious under 35 U.S.C. § 103(a) in view of **Shteyn** and **Carmel**.

Ground 2b: Claim 4 is obvious under 35 U.S.C. § 103(a) in view of **Shteyn** and **Glaser**.

Ground 3a: Claims 1 and 4 are anticipated under 35 U.S.C. § 102(e) in view of **Hill**.

Ground 3b: Claim 4 is obvious under 35 U.S.C. § 103(a) in view of **Hill** and **Carmel**.

Ground 3c: Claim 4 is obvious under 35 U.S.C. § 103(a) in view of **Hill** and **Glaser**.

Ground 4: Claims 1 and 4 are obvious under 35 U.S.C. § 103(a) in view of **Shteyn** and **Hill**.

Ground 5a: Withdrawn

Ground 5b: Withdrawn

Ground 5c: Withdrawn

Ground 6a: Claims 1 and 4 are anticipated under 35 U.S.C. § 102(e) in view of **Carmel**.

Ground 6b: Claims 1 and 4 are obvious under 35 U.S.C. § 103(a) in view of **Carmel** and **Hill**.

Ground 7a: Withdrawn

Ground 7b: Withdrawn

Ground 7c: Withdrawn

Ground 8: Withdrawn

Ground 9a: Claim 1 is anticipated under 35 U.S.C. § 102(e) in view of **Imai**.

Ground 9b: Claim 4 is obvious under 35 U.S.C. § 103(a) in view of **Imai** and **Carmel**.

Ground 9c: Claim 4 is obvious under 35 U.S.C. § 103(a) in view of **Imai** and **Glaser**.

Relevant Statutes

The following is a quotation of the appropriate paragraphs of pre-AIA 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

The following is a quotation of pre-AIA 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claim Rejections

<u>Ground 1: Shteyn</u>

Claims 1 and 4 are anticipated under 35 U.S.C. § 102(e) in view of **Shteyn**.

Regarding claim 1, **Shteyn** teaches **a media player** (see disclosure of a multi-purpose computing device, such as a PC or a set top box, col. 4, lines 38-44 et seq.) for receiving **an audio or video program** as claimed, **the program comprising media data elements** (see disclosure of a content file that is split into multiple parts, col. 1, line 65; see also disclosure regarding the nature of the file parts, col. 2, lines 25-38 et seq.), from **a media source over an Internet protocol network** (see disclosure that the first file segment is downloaded for play out by communicating with a remote server via techniques well known in the art, col. 3, lines 7-13 et seq.; see also disclosure that the

segments are accessed over the Internet, col. 3, lines 47-50 et seq.), and **playing the program for a user of the media player** (see disclosure that the rendering of the first segment is started by forwarding buffered content of the first segment to a decoding/playing module via techniques well known in the art, col. 3, lines 8-13 et seq.), wherein **each of the media data elements is associated with a serial number** (see disclosure that upon completion of the first segment, the second segment is passed on from the buffer to the decoding/playing module, through the use of e.g., a linked list wherein each segment has content data and a pointer to the next segment, col. 3, lines 31-36 et seq.; see also disclosure of an XML code fragment associated with a content file including a sequence of serial numbers embodied in metatags representing a sequence of file part numbers such as part1, part2, etc., Fig. 2:

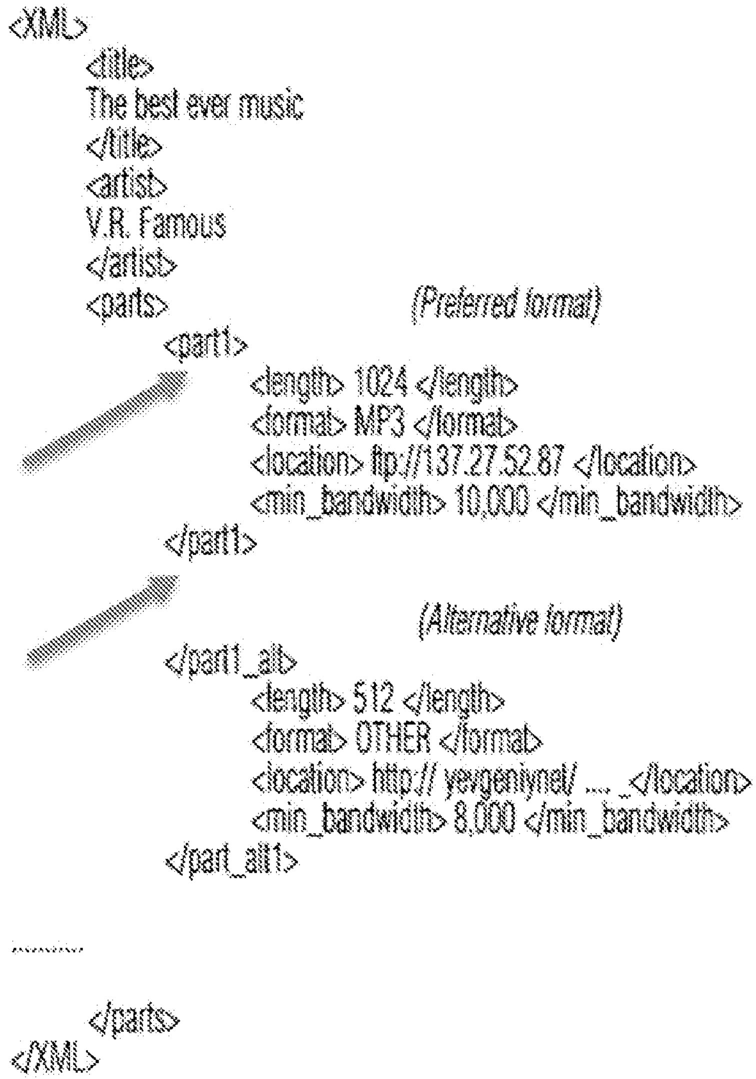


FIG. 2

),

comprising:

a) **a processor** (see disclosure that the invention relates to content and/or control communications between multiple computer systems, or to such

communication between computer systems and consumer devices, col. 1, lines 6-9; see also disclosure that the client device can be embodied in a PC, col. 4, lines 38-44 et seq.; the Office notes that computer systems/PCs inherently include a processor);

b) **a memory** (see disclosure of the use of memory buffers to contain buffered file segments waiting to be played out, col. 4, lines 28-31; see also Abstract; see also col. 3, lines 8-18 et seq.);

c) **a connection to the network** (see disclosure of the downloading and play out of content file parts via the Internet, col. 3, lines 44-61 et seq.; see also preamble of claim 12, col. 6, lines 41-43); and

d) **media player software** (see disclosure of decoding/playing module, col. 3, lines 8-13 et seq.) comprising:

i) instructions to cause the media player to **request from the media source a predetermined number of media data elements** (see disclosure that the client downloads content files parts to the buffer prior to play out, col. 3, lines 3-16 et seq.; see also disclosure that the client may make its own decisions regarding how many parts to download before the start of play out, col. 2, lines 23-24 et seq.);

ii) instructions to cause the media player to **receive media data elements sent to the media player by the media source and store the media data elements in the memory** (see disclosure that file segments are downloaded at the client and stored in the buffer while the previous file segment is being played out, and that the downloaded files can be buffered in a sequence or linked list of buffers, col. 3, lines 3-30 et seq.);

iii) instructions to **implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer** (see numerous disclosures describing the management of the file buffer memory, such as the disclosure that the file segments are downloaded for play out, such as by utilizing Java 2.0's set of standard classes that enable retrieving remote files into a buffer or as a stream, and that file segments are downloaded at the client and stored in the buffer while the previous file segment is being played out, and that the downloaded files can be buffered in a sequence or linked list of buffers, col. 3, lines 3-30 et seq.; see also disclosure that the client downloads control information for a desired content file that enables

retrieving and playing out of the segmented file; see also disclosure that the control information can be coded in XML, such that the XML code is parsed at the client, col. 2, line 57 through col. 3, line 2 et seq.; see also disclosure that each file part is defined in the XML control information with XML tags defining the sequential serial number such as'part1', 'part2', etc., Fig. 2; see also disclosure that the client may make its own decisions regarding how many parts to download before the start of play out, col. 2, lines 23-24 et seq.);

iv) instructions to cause the media player to **play media data elements sequentially from the player buffer** (see flowchart of the download and play out process, Fig. 1, as well as a description of the process at col. 2, line 53 through col. 3, lines 43:

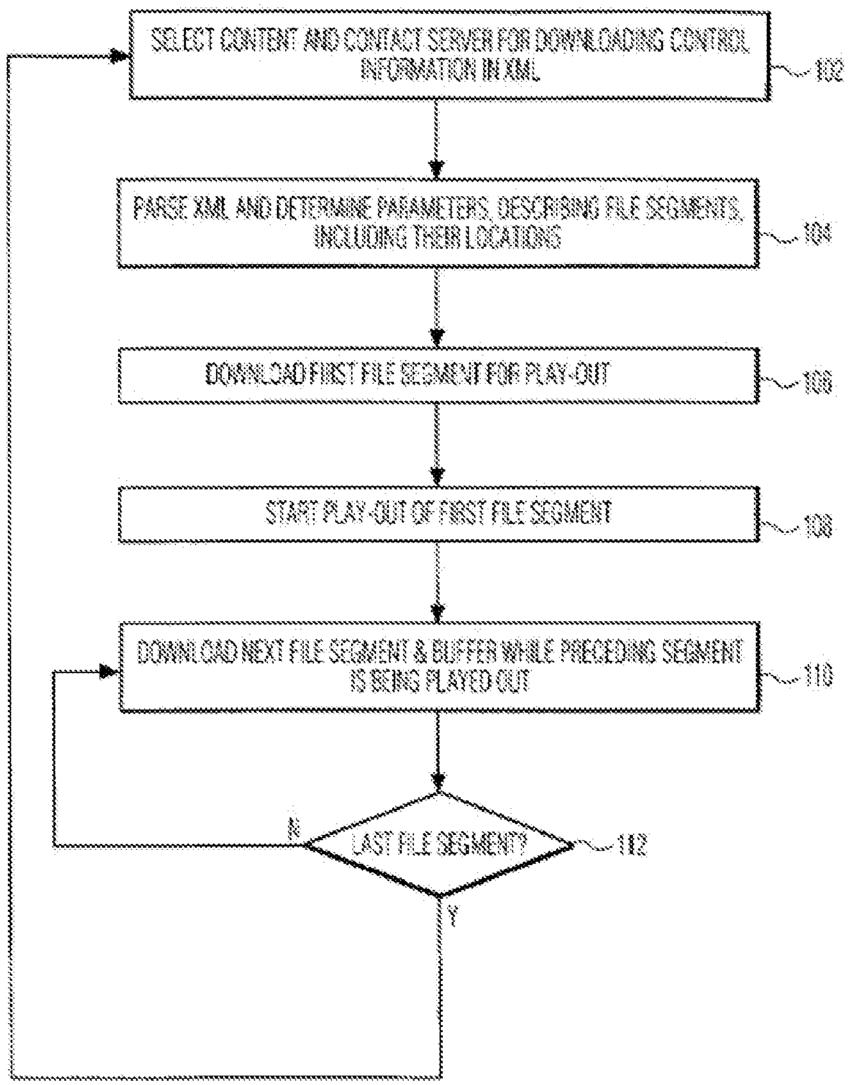


FIG. 1

; see particularly the loop of steps 110-112); and

v) instructions to cause the media player to **transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-**

determined number of media data elements in the player buffer until the last media data element comprising the program has been received (see disclosure that the client downloads control information for a desired content file that enables retrieving and playing out of the segmented file; see also disclosure that the control information can be coded in XML, such that the XML code is parsed at the client, col. 2, line 57 through col. 3, line 2 et seq.; see also disclosure that each file part is defined in the XML control information with XML tags defining the sequential serial number such as'part1', 'part2', etc., Fig. 2; see also disclosure that the client downloads the next file segment while the previous file segment is being played out, and that the downloaded file segments are buffered in a sequence of linked list of buffers, col. 3, lines 3-30 et seq.; see also disclosure that the client may make its own decisions regarding how many parts to download before the start of the play out, col. 2, lines 23-24 et seq.; see also steps 110-112 of the flowchart of Fig. 1, illustrating the process of repeatedly downloading file segments while playing out previously downloaded file segments, until the last file segment has been downloaded).

Regarding claim 4, **Shteyn** additionally teaches the media player of claim 1, wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to **receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player** (see disclosure that an object of the invention is to allow for low or negligible play-out latency by splitting a content file into multiple parts, each of which requires a relatively short download time, col. 1, lines 62-66 et seq.; see also disclosure that the client initially downloads control information associated with a desired content file, said control information including information regarding format, length, required bandwidth, location on the server, etc., col. 2, lines 3-7, 20-24, and 57-63 et seq.; see also disclosure that the control information also includes information regarding alternative content parts/segments, col. 3, lines 25-29 and col. 3, lines 44-53 et seq.; see also disclosure that the client can automatically select the format compatible with the client's play-out capabilities, col. 3, lines 55-56 et seq.; see also an illustration of control information including information regarding alternate sources for content segments having different minimum bandwidth requirements, Fig. 2:

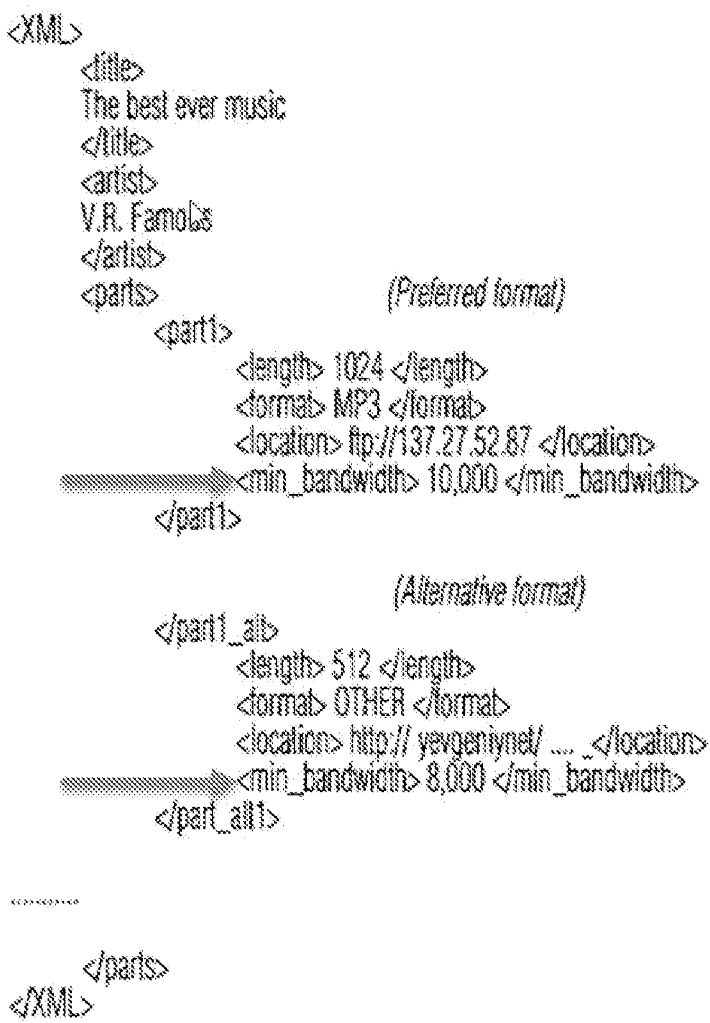


FIG. 2

; see also disclosure that during operation, the client application could select a next segment in a different format for the same content to adapt to changing circumstances, e.g., lower bandwidth due to network congestion, thus achieving the stated object of the invention of minimizing latency by ensuring that the segments are

downloaded at a rate that exceeds the rate at which they are played out, col. 4, lines 20-23 et seq.).

Ground 2a: Shteyn and Carmel

Claim 4 is obvious under 35 U.S.C. § 103(a) in view of **Shteyn** and **Carmel**.

Regarding claim 4, **Shteyn** teaches the media player of claim 1 (see rejection of claim 1, Ground 1, above), wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to **receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player** (see disclosure that an object of the invention is to allow for low or negligible play-out latency by splitting a content file into multiple parts, each of which requires a relatively short download time, col. 1, lines 62-66 et seq.; see also disclosure that the client initially downloads control information associated with a desired content file, said control information including information regarding format, length, required bandwidth, location on the server, etc., col. 2, lines 3-7, 20-24, and 57-63 et seq.; see also disclosure that the control information also includes information

regarding alternative content parts/segments, col. 3, lines 25-29 and col. 3, lines 44-53 et seq.; see also disclosure that the client can automatically select the format compatible with the client's play-out capabilities, col. 3, lines 55-56 et seq.; see also an illustration of control information including information regarding alternate sources for content segments having different minimum bandwidth requirements, Fig. 2:

```
&lt;XML&gt;
    &lt;title&gt;
    The best ever music
    &lt;/title&gt;
    &lt;artist&gt;
    V.R. Famous
    &lt;/artist&gt;
    &lt;parts&gt;                    (Preferred format)
        &lt;part1&gt;
            &lt;length&gt; 1024 &lt;/length&gt;
            &lt;format&gt; MP3 &lt;/format&gt;
            &lt;location&gt; ftp://137.27.52.87 &lt;/location&gt;
            &lt;min_bandwidth&gt; 10,000 &lt;/min_bandwidth&gt;
        &lt;/part1&gt;

                               (Alternative format)
        &lt;/part1_alt&gt;
            &lt;length&gt; 512 &lt;/length&gt;
            &lt;format&gt; OTHER &lt;/format&gt;
            &lt;location&gt; http:// yevgeniynet/ ... _&lt;/location&gt;
            &lt;min_bandwidth&gt; 8,000 &lt;/min_bandwidth&gt;
        &lt;/part_alt1&gt;

..........

    &lt;/parts&gt;
&lt;/XML&gt;
```

FIG. 2

; see also disclosure that during operation, the client application could select a next segment in a different format for the same content to adapt to changing circumstances, e.g., lower bandwidth due to network congestion, thus achieving the stated object of the invention of minimizing latency by ensuring that the segments are downloaded at a rate that exceeds the rate at which they are played out, col. 4, lines 20-23 et seq.).

To the extent that one could argue that **Shteyn** fails to explicitly teach **receipt of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player**, **Carmel** also teaches this limitation[1].

Specifically, **Carmel** teaches that with respect to a system for real-time multimedia broadcasting over a network, during normal operation the client computer monitors the downloading of data from the server in order to determine the amount of time required to convey each media slice and to verify that the slices are conveyed at a sufficient rate. When the data stream comprises multimedia data, the data rate should

[1] This conclusion is consistent with the Final Written Decision on Remand by the PTAB in IPR2016-01238, 16 July 2020.

be generally equal to or faster than the rate at which the data are generated at the transmitting computer. See col. 2, lines 51-59.

Carmel also discloses a lag-recovery mechanism whereby if a lag is detected in the data transmission/reception, steps are taken to increase the data transmission or reception rate, such as by adjusting the data compression level. See col. 7, lines 35-49 and col. 11, lines 9-22.

A POSITA would have found it obvious to combine the teachings of **Carmel** and **Shteyn** because the combination is merely a matter of applying a known technique to a known device for improvement to yield predictable results.

Shteyn and **Carmel** disclose similar client-pull systems, including requesting data by serial number using a control or index file. They also both teach the client's ability to select different quality or bandwidth files from the index file depending on network bandwidth. Compare **Shteyn**, at 3:44-53, 3:55-56, 4:20-26, Fig. 2, claims 1, 12, with **Carmel**, at 2:24-27, 3:5-9, 7:24-27, 8:42-9:5, Fig. 3D.

A POSITA under KSR at the time of the purported invention was made would have been motivated to combine the teachings of **Carmel** and **Shteyn** given the similarity of the **Shteyn** and **Carmel** systems and a POSITA would have been motivated to review other client-pull systems to further efficiency and enhance **Shteyn's**

functionality. This includes, for instance, implementing **Carmel's** sending faster than playback rate in **Shteyn**.

A POSITA would have been further motivated to implement such function and would have had a reasonable expectation of success in doing so considering **Shteyn's** existing server and client functionality. **Shteyn** has client buffers (or linked list of buffers) such that the predetermined segments could be received faster than the playback rate. **Shteyn**, at Abstract, 1:6-14, 3:31-36, 3:14-18. Moreover, **Shteyn** has an existing system, just like **Carmel**, to select the appropriate segment depending on network bandwidth. **Shteyn**, at 3:44-53, 3:55-56, 4:20-26, Fig. 2, claims 1, 12. Further, **Shteyn** teaches that the server already partitions media into segments and that the invention could be used for real-time communications such as broadcasts to a plurality of users. Thus, **Shteyn** could implement **Carmel's** sending faster than playback rate seamlessly.

Accordingly, a POSITA would have been motivated to combine **Shteyn** and **Carmel**.

<u>Ground 2b: Shteyn and Glaser</u>

Claim 4 is obvious under 35 U.S.C. § 103(a) in view of **Shteyn** and **Glaser**.

Regarding claim 4, **Shteyn** teaches the media player of claim 1 (see rejection of claim 1, Ground 1, above), wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to **receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player** (see disclosure that an object of the invention is to allow for low or negligible play-out latency by splitting a content file into multiple parts, each of which requires a relatively short download time, col. 1, lines 62-66 et seq.; see also disclosure that the client initially downloads control information associated with a desired content file, said control information including information regarding format, length, required bandwidth, location on the server, etc., col. 2, lines 3-7, 20-24, and 57-63 et seq.; see also disclosure that the control information also includes information regarding alternative content parts/segments, col. 3, lines 25-29 and col. 3, lines 44-53 et seq.; see also disclosure that the client can automatically select the format compatible with the client's play-out capabilities, col. 3, lines 55-56 et seq.; see also an illustration of control information including information regarding alternate sources for content segments having different minimum bandwidth requirements, Fig. 2:

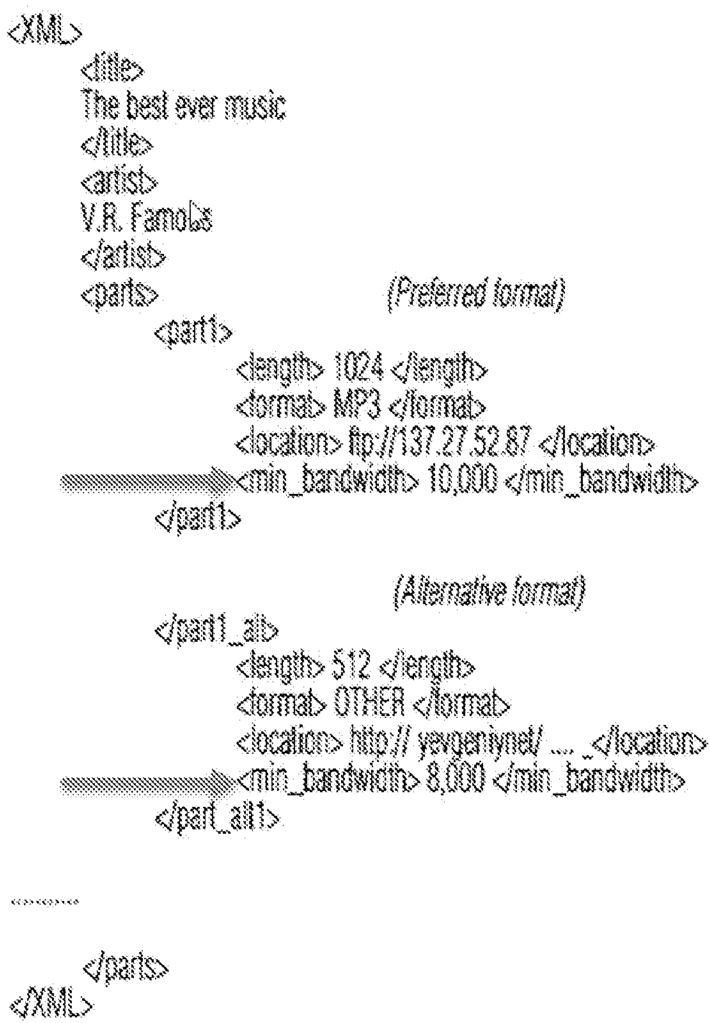


FIG. 2

; see also disclosure that during operation, the client application could select a next segment in a different format for the same content to adapt to changing circumstances, e.g., lower bandwidth due to network congestion, thus achieving the stated object of the invention of minimizing latency by ensuring that the segments are

downloaded at a rate that exceeds the rate at which they are played out, col. 4, lines 20-23 et seq.).

To the extent that one could argue that **Shteyn** fails to explicitly teach **receipt of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player**, **Glaser** also teaches this limitation.

Specifically, **Glaser** teaches that with respect to an audio-on-demand communication system, the client buffer receives data at a rate faster than the playback rate to maintain a predetermined threshold. For instance, at col. 22, lines 57-59, **Glaser** discloses that since the normal quality data blocks are transmitted at greater than real time, the buffer begins to refill and approach maximum capacity. At col. 21, lines 57-60, it is disclosed that when the subscriber PC determines that the buffers are near maximum capacity (e.g., above 85% of capacity), this indicates that the normal quality data is being transferred in real time or greater than real time. At col. 20, lines 8-16, it is disclosed that the audio-on-demand system allows for greater than real time delivery of audio data to the subscriber PC in many cases.

A POSITA would have been motivated to combine **Glaser** and **Shteyn**. A POSITA would have been motivated to implement **Glaser's** sending faster than the

playback rate in **Shteyn** to enhance **Shteyn's** functionality, including real-time playback. **Shteyn** at Abstract, 1:62-66. A POSITA would have been further motivated to implement such function considering **Shteyn's** existing server structure and buffering ability to store the data sent faster than the playback rate. **Shteyn** at Abstract, 1:62-66, 3:14-18, 3:31-36, 4:20-31. Moreover, **Shteyn** has an existing system, just like **Glaser**, to receive the appropriate quality data. Compare **Shteyn**, at 3:44-53, 3:57-61, 4:20-26, Fig. 1, with **Glaser**, at 21:57-63, 22:41-59.

Accordingly, a POSITA would have been motivated to combine **Shteyn** and **Glaser**.

<u>Ground 3a: Hill</u>

Claims 1 and 4 are anticipated under 35 U.S.C. § 102(e) in view of **Hill**.

Regarding claim 1, **Hill** teaches **a media player** (see disclosure that the invention relates to audio/video playback devices, col. 1, lines 7-11 et seq.) for receiving **an audio or video program** as claimed, **the program comprising media data elements** (see disclosure that movie data is composed of synchronized audio and video data broken into fundamental units called frames, col. 3, lines 37-39 et seq.; see also disclosure of

clips, which are a sequence of frames, col. 3, lines 46-47 et seq.), from **a media source over an Internet protocol network** (see disclosure that clips may be stored in different locations within the digital network environment, that movie data contains a number of clips stored in separate sources, and that a source may be a site on the Internet, col. 3, lines 48-56 et seq.), and **playing the program for a user of the media player** (see disclosure that the user utilizes the workstation to play the movie data, col. 3, lines 28-29 et seq.), wherein **each of the media data elements is associated with a serial number** (see disclosure that frames are requested with respect to a frame number (i.e., serial number), col. 5, line 59 through col. 6, line 16 and col. 7, lines 39-50 et seq.; see also disclosure of the use of frame numbers (i.e., serial numbers), for identifying specific frames, Figs. 6B and 7:

FIG. 6B

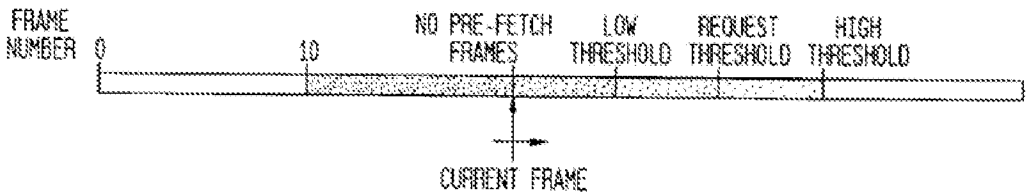
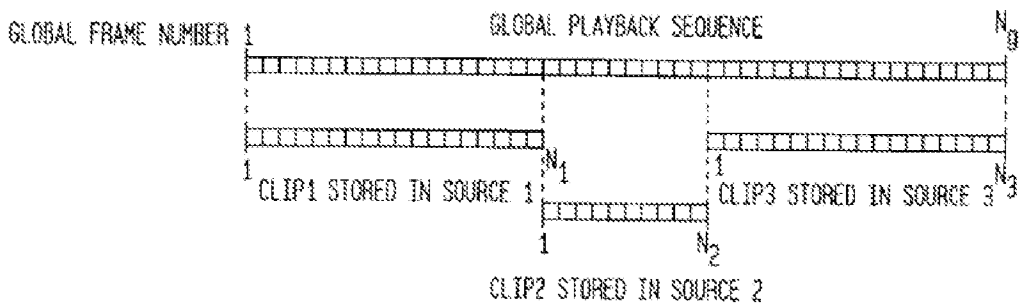


FIG. 7



),

comprising:

a) **a processor** (see disclosure of a workstation that may be a conventional personal computer, a high performance workstation, or dedicated hardware specially suited for audio/video playback, col. 3, lines 28-32 et seq.; see also control unit 114, Fig. 1; the Office notes that computer systems/personal computers inherently include a processor);

b) **a memory** (see disclosure that the buffer manager stores frames in local digital memory, such as random access memory (RAM) or a local disk drive, col. 4, lines 7-10 et seq.);

c) **a connection to the network** (see disclosure of a workstation connected to a digital network, Abstract; see also disclosure that clips may be stored in different locations within the digital network environment, that movie data contains a number of clips stored in separate sources, and that a source may be a site on the Internet, col. 3, lines 48-56 et seq.); and

d) **media player software** (see disclosure of decompressor/renderer 110, Fig. 1; see also disclosure that the user utilizes the workstation to play movie data, and that the user views the movie on display 108 while issuing control information such as 'forward', 'backward', faster', etc., via a standard user interface, col. 3, lines 27-36 et seq.) comprising:

i) instructions to cause the media player to **request from the media source a predetermined number of media data elements** (see disclosure that frames are requested from the source, col. 7, lines 12-17; col. 5, lines 52-54; col. 9, lines 40-44 et seq.);

ii) instructions to cause the media player to **receive media data elements sent to the media player by the media source and store the media data elements in the memory** (see disclosure of steps 308 through 322, Fig. 3, illustrating the process of requesting new frames from the

source, and storing the received frames in the buffer, col. 5, line 48 through col. 6, line 34 et seq.);

iii) instructions to **implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer** (see disclosure that the buffer manager uses the global frame number last requested as the starting point for determining which frame to request next, col. 10, lines 4-6 et seq.);

iv) instructions to cause the media player to **play media data elements sequentially from the player buffer** (see disclosure that a clip is a sequence of frames and that movie data of interest is composed of one or more clips, col. 3, lines 46-47; see also disclosure that the decompressor/renderer presents the audio and video to the display in correct synchronization, and that the control unit is responsible for feeding frames to the decompressor/renderer in the proper sequence, col. 3, line 65 through col. 4, line 6 et seq.); and

v) instructions to cause the media player to **transmit to the media source a request to send one or more media data elements, each identified by**

a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received (see disclosure that the buffer manager requests two new frames when the fill level is two frames less than the request threshold, one frame when the fill level is one frame less than the request threshold, and no frames otherwise, col. 9, lines 40-44 et seq.; see also disclosure that the request threshold represents the optimum number of future frames stored in the buffer at any given time, and is determined based on the characteristics of the particular system (e.g., I/O bandwidth, source latency, etc.), col. 6, lines 58-67 et seq.; see also disclosure that the buffer manager uses the global frame number last requested as the starting point for determining which frame to request next, col. 10, lines 4-6 et seq).

Regarding claim 4, **Hill** additionally teaches the media player of claim 1, wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to

receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player (see disclosure that if the buffer manager is more than one frame behind, it will request two new frames, but that no more than two frames will be requested in order to avoid overloading the sources, and that if the play rate is 24 frames per second, the sources will not be asked to supply more than 48 frames per second, col. 7, lines 11-24 et seq.).

<u>Ground 3b: Hill and Carmel</u>

Claim 4 is obvious under 35 U.S.C. § 103(a) in view of **Hill** and **Carmel**.

Regarding claim 4, **Hill** teaches the media player of claim 1 (see rejection of claim 1, Ground 3a, above), wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to **receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player** (see disclosure that if the buffer manager is more than one frame behind, it will request two new frames, but that no more than two frames will be requested in order to avoid overloading the sources, and that if the play rate is 24

frames per second, the sources will not be asked to supply more than 48 frames per second, col. 7, lines 11-24 et seq.).

To the extent that one could argue that **Hill** fails to explicitly teach **receipt of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player**, **Carmel** also teaches this limitation[2].

Specifically, **Carmel** teaches that with respect to a system for real-time multimedia broadcasting over a network, during normal operation the client computer monitors the downloading of data from the server in order to determine the amount of time required to convey each media slice and to verify that the slices are conveyed at a sufficient rate. When the data stream comprises multimedia data, the data rate should be generally equal to or faster than the rate at which the data are generated at the transmitting computer. See col. 2, lines 51-59.

Carmel also discloses a lag-recovery mechanism whereby if a lag is detected in the data transmission/reception, steps are taken to increase the data transmission or

[2] This conclusion is consistent with the Final Written Decision on Remand by the PTAB in IPR2016-01238, 16 July 2020.

reception rate, such as by adjusting the data compression level. See col. 7, lines 35-49 and col. 11, lines 9-22.

A POSITA would have found it obvious to combine the teachings of **Carmel** and **Hill** because the combination is merely a matter of applying a known technique to a known device for improvement to yield predictable results.

Hill and **Carmel** both teach video distribution systems over the Internet, including requesting data by serial number and to repeat such requests. Further, both maintain a record of the last serial number received. A POSITA under KSR at the time of the purported invention was made would have been motivated to combine the similar systems. For instance, a POSITA would have been motivated to implement **Carmel's** sending faster than playback rate to further enhance **Hill's** playback. **Hill**, at 6:58-67.

A POSITA would have been further motivated to implement such function and would have had a reasonable expectation of success in doing so considering **Hill's** existing functionality. **Hill** already discloses maintaining a request threshold number of frames based on system characteristics (**Hill** at 1:7-11, 6:58-67, 10:55-60) and the server sending 48 frames per second or twice the rate of the playback rate at 24 frames per second. **Hill** at 7:12-24. Moreover, **Hill** teaches that its client buffering system adapts to

data sources that vary in speed and response time. **Hill** at Abstract, 1:7-11. Thus, **Hill** could implement **Carmel's** sending faster than playback given the existing server and client structure in **Hill** and overlapping similarities discussed above.

Accordingly, a POSITA would have been motivated to combine **Hill** and **Carmel**.

<u>Ground 3c: Hill and Glaser</u>

Claim 4 is obvious under 35 U.S.C. § 103(a) in view of **Hill** and **Glaser**.

Regarding claim 4, **Hill** teaches the media player of claim 1 (see rejection of claim 1, Ground 3a, above), wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to **receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player** (see disclosure that if the buffer manager is more than one frame behind, it will request two new frames, but that no more than two frames will be requested in order to avoid overloading the sources, and that if the play rate is 24

frames per second, the sources will not be asked to supply more than 48 frames per second, col. 7, lines 11-24 et seq.).

To the extent that one could argue that **Hill** fails to explicitly teach **receipt of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player**, **Glaser** also teaches this limitation.

Specifically, **Glaser** teaches that with respect to an audio-on-demand communication system, the client buffer receives data at a rate faster than the playback rate to maintain a predetermined threshold. For instance, at col. 22, lines 57-59, **Glaser** discloses that since the normal quality data blocks are transmitted at greater than real time, the buffer begins to refill and approach maximum capacity. At col. 21, lines 57-60, it is disclosed that when the subscriber PC determines that the buffers are near maximum capacity (e.g., above 85% of capacity), this indicates that the normal quality data is being transferred in real time or greater than real time. At col. 20, lines 8-16, it is disclosed that the audio-on-demand system allows for greater than real time delivery of audio data to the subscriber PC in many cases.

A POSITA would have been motivated to combine **Glaser** with **Hill**. **Hill** and **Glaser** both teach video distribution systems over the Internet. **Glaser** transmits

multimedia data faster than the playback rate to maintain a client buffer threshold. **Glaser**, at 3:15-28, 3:31-36, 13:36-44, 16:2-10, 21:57-63, 22:41-59. A POSITA under KSR would have been motivated to implement **Glaser's** sending data faster than the playback rate in **Hill** to enhance **Hill's** functionality, including playback. **Hill**, at Abstract, 6:58-67.

A POSITA would have been further motivated to implement such function considering **Hill's** existing functionality. **Hill** already teaches maintaining a threshold of data and sending at twice the rate of the playback rate. Moreover, **Hill** teaches that its client buffering system adapts to data sources that vary in speed and response time. **Hill** at Abstract, 1:7-11. Thus, **Hill** could buffer the data sent by **Glaser** faster than the playback rate.

Accordingly, a POSITA would have been motivated to combine **Hill** and **Glaser**.

Ground 4: Shteyn and Hill

Claims 1 and 4 are obvious under 35 U.S.C. § 103(a) in view of **Shteyn** and **Hill**.

Regarding claim 1, **Shteyn** teaches **a media player** (see disclosure of a multi-purpose computing device, such as a PC or a set top box, col. 4, lines 38-44 et seq.) for

receiving **an audio or video program** as claimed, **the program comprising media data elements** (see disclosure of a content file that is split into multiple parts, col. 1, line 65; see also disclosure regarding the nature of the file parts, col. 2, lines 25-38 et seq.), from **a media source over an Internet protocol network** (see disclosure that the first file segment is downloaded for play out by communicating with a remote server via techniques well known in the art, col. 3, lines 7-13 et seq.; see also disclosure that the segments are accessed over the Internet, col. 3, lines 47-50 et seq.), and **playing the program for a user of the media player** (see disclosure that the rendering of the first segment is started by forwarding buffered content of the first segment to a decoding/playing module via techniques well known in the art, col. 3, lines 8-13 et seq.), wherein **each of the media data elements is associated with a serial number** (see disclosure that upon completion of the first segment, the second segment is passed on from the buffer to the decoding/playing module, through the use of e.g., a linked list wherein each segment has content data and a pointer to the next segment, col. 3, lines 31-36 et seq.; see also disclosure of an XML code fragment associated with a content file including a sequence of serial numbers embodied in metatags representing a sequence of file part numbers such as part1, part2, etc., Fig. 2:

```
<XML>
      <title>
      The best ever music
      </title>
      <artist>
      V.R. Famous
      </artist>
      <parts>                    (Preferred format)
            <part1>
                  <length> 1024 </length>
                  <format> MP3 </format>
                  <location> ftp://137.27.52.87 </location>
                  <min_bandwidth> 10,000 </min_bandwidth>
            </part1>

                                 (Alternative format)
            </part1_alt>
                  <length> 512 </length>
                  <format> OTHER </format>
                  <location> http:// yevgeniynet/ ..... _</location>
                  <min_bandwidth> 8,000 </min_bandwidth>
            </part_alt1>

......

      </parts>
</XML>
```

FIG. 2

),

comprising:

a) **a processor** (see disclosure that the invention relates to content and/or control communications between multiple computer systems, or to such

communication between computer systems and consumer devices, col. 1, lines 6-9; see also disclosure that the client device can be embodied in a PC, col. 4, lines 38-44 et seq.; the Office notes that computer systems/PCs inherently include a processor);

b) **a memory** (see disclosure of the use of memory buffers to contain buffered file segments waiting to be played out, col. 4, lines 28-31; see also Abstract; see also col. 3, lines 8-18 et seq.);

c) **a connection to the network** (see disclosure of the downloading and play out of content file parts via the Internet, col. 3, lines 44-61 et seq.; see also preamble of claim 12, col. 6, lines 41-43); and

d) **media player software** (see disclosure of decoding/playing module, col. 3, lines 8-13 et seq.) comprising:

i) instructions to cause the media player to **request from the media source a predetermined number of media data elements** (see disclosure that the client downloads content files parts to the buffer prior to play out, col. 3, lines 3-16 et seq.; see also disclosure that the client may make its own decisions regarding how many parts to download before the start of play out, col. 2, lines 23-24 et seq.);

ii) instructions to cause the media player to **receive media data elements sent to the media player by the media source and store the media data elements in the memory** (see disclosure that file segments are downloaded at the client and stored in the buffer while the previous file segment is being played out, and that the downloaded files can be buffered in a sequence or linked list of buffers, col. 3, lines 3-30 et seq.);

iii) instructions to **implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer** (see numerous disclosures describing the management of the file buffer memory, such as the disclosure that the file segments are downloaded for play out, such as by utilizing Java 2.0's set of standard classes that enable retrieving remote files into a buffer or as a stream, and that file segments are downloaded at the client and stored in the buffer while the previous file segment is being played out, and that the downloaded files can be buffered in a sequence or linked list of buffers, col. 3, lines 3-30 et seq.; see also disclosure that the client downloads control information for a desired content file that enables

retrieving and playing out of the segmented file; see also disclosure that the control information can be coded in XML, such that the XML code is parsed at the client, col. 2, line 57 through col. 3, line 2 et seq.; see also disclosure that each file part is defined in the XML control information with XML tags defining the sequential serial number such as'part1', 'part2', etc., Fig. 2; see also disclosure that the client may make its own decisions regarding how many parts to download before the start of play out, col. 2, lines 23-24 et seq.);

iv) instructions to cause the media player to **play media data elements sequentially from the player buffer** (see flowchart of the download and play out process, Fig. 1, as well as a description of the process at col. 2, line 53 through col. 3, lines 43:

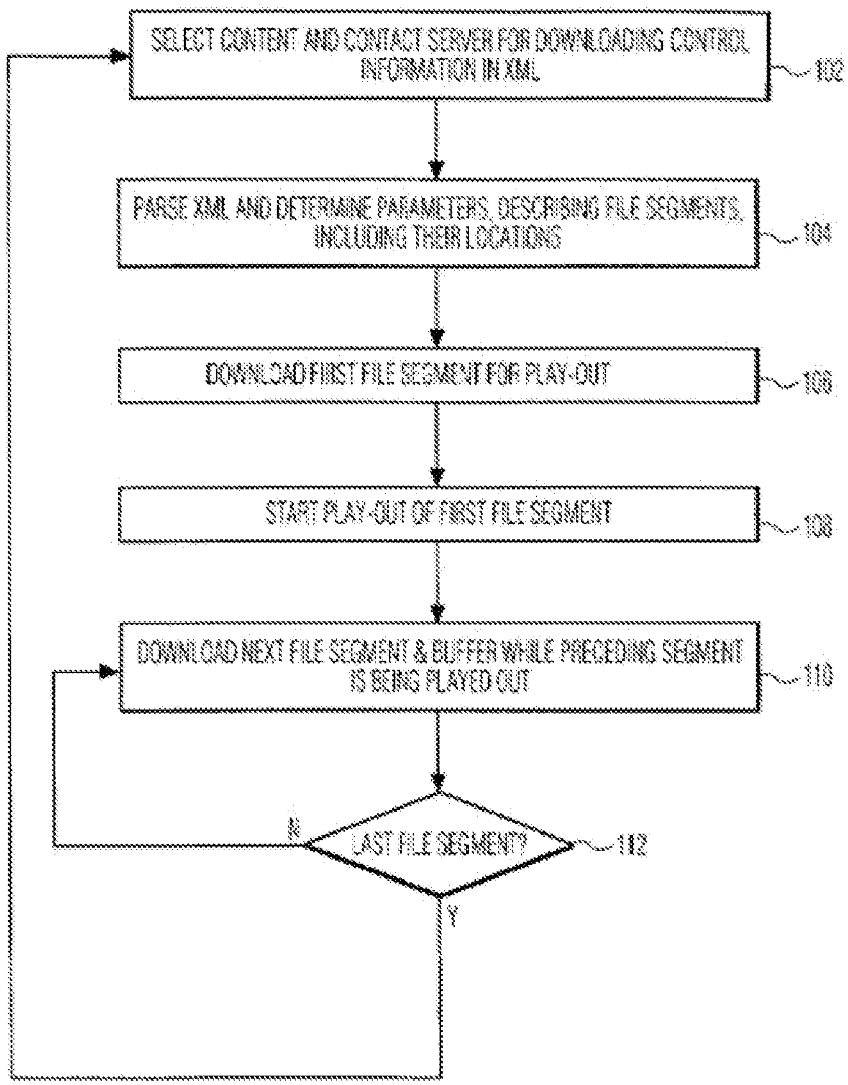


FIG. 1

; see particularly the loop of steps 110-112); and

v) instructions to cause the media player to **transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-**

determined number of media data elements in the player buffer until the last media data element comprising the program has been received (see disclosure that the client downloads control information for a desired content file that enables retrieving and playing out of the segmented file; see also disclosure that the control information can be coded in XML, such that the XML code is parsed at the client, col. 2, line 57 through col. 3, line 2 et seq.; see also disclosure that each file part is defined in the XML control information with XML tags defining the sequential serial number such as'part1', 'part2', etc., Fig. 2; see also disclosure that the client downloads the next file segment while the previous file segment is being played out, and that the downloaded file segments are buffered in a sequence of linked list of buffers, col. 3, lines 3-30 et seq.; see also disclosure that the client may make its own decisions regarding how many parts to download before the start of the play out, col. 2, lines 23-24 et seq.; see also steps 110-112 of the flowchart of Fig. 1, illustrating the process of repeatedly downloading file segments while playing out previously downloaded file segments, until the last file segment has been downloaded).

To the extent that one could argue that **Shteyn** fails to explicitly teach instructions to cause the media player to **transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received**, **Hill** also teaches this limitation.

Specifically, **Hill** teaches that with respect to a high performance player for distributed, time-based media, the media player software comprises instructions to cause the media player to **transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received** (see disclosure that the buffer manager requests two new frames when the fill level is two frames less than the request threshold, one frame when the fill level is one frame less than the request threshold, and no frames otherwise, col. 9, lines 40-44 et seq.; see also disclosure that the request threshold represents the optimum number of future frames stored in the buffer at any given time, and is determined based on the characteristics of the particular

system (e.g., I/O bandwidth, source latency, etc.), col. 6, lines 58-67 et seq.; see also disclosure that the buffer manager uses the global frame number last requested as the starting point for determining which frame to request next, col. 10, lines 4-6 et seq).

A POSITA would have found it obvious to combine the teachings of **Hill** and **Shteyn** because the combination is merely a matter of applying a known technique to a known device for improvement to yield predictable results.

Shteyn and **Hill** both teach video distribution systems over the Internet, including requesting data by serial number and to repeat such requests. A POSITA at the time of the purported invention was made would have been motivated to combine the similar systems and enhance **Shteyn's** functionality, including real-time playback. This includes, for instance, implementing **Hill's** buffer threshold. **Hill**, at 6:58-67, 7:1-2, 7:12-24, 9:40-44, 10:55-60, claims 8-9.

A POSITA under KSR would have been further motivated to implement such a function and would have had a reasonable expectation of success in doing so considering **Shteyn's** existing functionality. **Shteyn** already teaches buffering and real-time playback of sequential media segments. **Shteyn**, at Abstract, at 1:62-2:3, 3:3-36, 3:63-4:3, Fig. 1. Moreover, this existing buffering (linked list) enables **Shteyn** to store the received predetermined data faster than the playback rate. **Shteyn** at Abstract, 1:62-

66, 3:14-18, 3:31-36, 4:20-31. Thus, **Shteyn** could implement **Hill's** buffer threshold seamlessly.

Accordingly, a POSITA would have been motivated to combine **Shteyn** and **Hill**.

Regarding claim 4, **Shteyn** additionally teaches the media player of claim 1, wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to **receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player** (see disclosure that an object of the invention is to allow for low or negligible play-out latency by splitting a content file into multiple parts, each of which requires a relatively short download time, col. 1, lines 62-66 et seq.; see also disclosure that the client initially downloads control information associated with a desired content file, said control information including information regarding format, length, required bandwidth, location on the server, etc., col. 2, lines 3-7, 20-24, and 57-63 et seq.; see also disclosure that the control information also includes information regarding alternative content parts/segments, col. 3, lines 25-29 and col. 3, lines 44-53 et seq.; see also disclosure that the client can automatically select the format compatible with the client's play-out

capabilities, col. 3, lines 55-56 et seq.; see also an illustration of control information including information regarding alternate sources for content segments having different minimum bandwidth requirements, Fig. 2:


```
<XML>
    <title>
    The best ever music
    </title>
    <artist>
    V.R. Famous
    </artist>
    <parts>                     (Preferred format)
        <part1>
            <length> 1024 </length>
            <format> MP3 </format>
            <location> ftp://137.27.52.87 </location>
            <min_bandwidth> 10,000 </min_bandwidth>
        </part1>

                                (Alternative format)
        </part1_alt>
            <length> 512 </length>
            <format> OTHER </format>
            <location> http:// yevgeniynet/ .... _</location>
            <min_bandwidth> 8,000 </min_bandwidth>
        </part_alt1>

..........

    </parts>
</XML>
```


FIG. 2

; see also disclosure that during operation, the client application could select a next segment in a different format for the same content to adapt to changing circumstances, e.g., lower bandwidth due to network congestion, thus achieving the stated object of the invention of minimizing latency by ensuring that the segments are downloaded at a rate that exceeds the rate at which they are played out, col. 4, lines 20-23 et seq.).

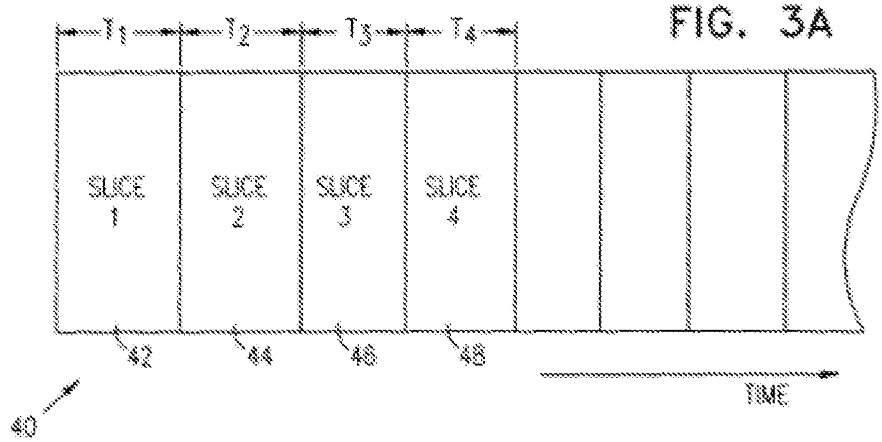
Ground 6a: Carmel

Claims 1 and 4 are anticipated under 35 U.S.C. § 102(e) in view of **Carmel**.

Regarding claim 1, **Carmel** teaches **a media player** (see disclosure that clients typically need only add a Java applet or plug-in to their existing web browser in order to receive and play back a broadcast, col. 7, lines 12-17 et seq.) for receiving **an audio or video program** as claimed, **the program comprising media data elements** (see disclosure that the data stream is divided into a sequence of segments or slices, preferably time slices, col. 2, lines 4-5 et seq.), from **a media source over an Internet protocol network** (see disclosure that a transmitting computer generates a data stream and broadcasts the data stream via a network server to a plurality of clients, and that

the clients download the data stream from the server, preferably using an Internet protocol, col. 2, lines 1-15 et seq.; see also disclosure of broadcasting video data over the Internet, col. 1, lines 16-22 et seq.), and **playing the program for a user of the media player** (see disclosure that the one or more client computers decode the sequence and play back the data stream responsive to the indices thereof, col. 5, lines 22-25 et seq.), wherein **each of the media data elements is associated with a serial number** (see disclosure that each slice of the data stream is assigned a respective slice index, col. 2, lines 4-7 and Fig. 3A:

FIG. 3A



),

comprising:

a) **a processor** (see disclosure that clients comprise conventional personal computers or workstations, col. 6, lines 38-40 et seq.; the Office notes that personal computers and workstations inherently include processors);

b) **a memory** (see disclosure that clients comprise conventional personal computers or workstations, col. 6, lines 38-40 et seq.; the Office notes that personal computers and workstations inherently include memory);

c) **a connection to the network** (see disclosure that a transmitting computer generates a data stream and broadcasts the data stream via a network server to a plurality of clients, and that the clients download the data stream from the server, preferably using an Internet protocol, col. 2, lines 1-15 et seq.; see also disclosure of broadcasting video data over the Internet, col. 1, lines 16-22 et seq.); and

d) **media player software** (see disclosure that clients typically need only add a Java applet or plug-in to their existing web browser in order to receive and play back a broadcast, col. 7, lines 12-17 et seq.) comprising:

i) instructions to cause the media player to **request from the media source a predetermined number of media data elements** (see disclosure that the 'Fetch segment' module fetches the first segment and stores it in BUFF_A by making a request for URL(1), URL(2), and so on, until URL(A1), followed by execution of the 'Fetch next segment-and display present-segment' module which fetches the second segment and stores it in BUFF_B while simultaneously a client typically opens

one or two HTTP links over which files 42, 44, 46, etc. are downloaded in successive alternation, col. 10, lines 36-40 et seq.);

ii) instructions to cause the media player to **receive media data elements sent to the media player by the media source and store the media data elements in the memory** (see disclosure that client computer 30 downloads and decodes files 42, 44, 46, etc., and reconstructs the multimedia data for output to the user through the use of timestamps, col. 10, lines 35-54 et seq.; the Office notes that decoding and reconstructing/synchronizing the downloaded data would require said downloaded data first be stored in a memory);

iii) instructions to **implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer** (see disclosure that the operation of client 30 is controlled by a Java applet that includes facilities for carrying out the steps shown in Fig. 6A, col. 10, lines 27-34 et seq.; see also disclosure that the client selects an appropriate starting slice of data stream 40 and begins to download and decompress files 42, 44, 46, etc. in successive alternation and that client 30 reconstructs

and outputs the multimedia data, col. 10, lines 35-50 et seq.; see also disclosure that the division of the data stream into slices and the inclusion of the slice indices in the data stream is used by the clients to maintain synchronization, col. 2, lines 17-21 et seq.);

iv) instructions to cause the media player to **play media data elements sequentially from the player buffer** (see disclosure that the one or more client computers decode the sequence and play back the data stream responsive to the indices thereof, col. 5, lines 22-25 et seq.); and

v) instructions to cause the media player to **transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received** (see disclosure that the client selects an appropriate starting slice of data stream 40 and begins to download and decompress files 42, 44, 46, etc. in successive alternation and that client 30 reconstructs and outputs the multimedia data, col. 10, lines 35-50 et seq.; see also disclosure that client 30 monitors the time codes as the files are

received in order to ensure that the transmission or reception is 'keeping up' with the input of data to the computer, col. 7, lines 36-40 et seq.).

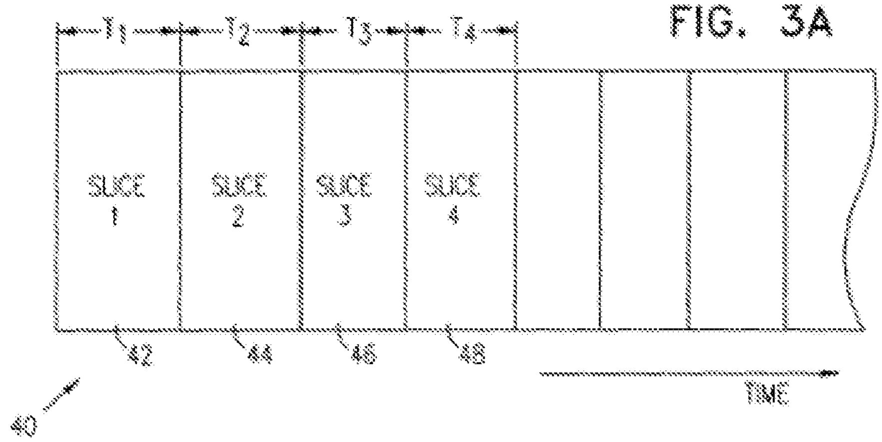
Regarding claim 4, **Carmel** additionally teaches the media player of claim 1, wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to **receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player** (see disclosure that with respect to a system for real-time multimedia broadcasting over a network, during normal operation the client computer monitors the downloading of data from the server in order to determine the amount of time required to convey each media slice and to verify that the slices are conveyed at a sufficient rate, and that when the data stream comprises multimedia data, the data rate should be generally equal to or faster than the rate at which the data are generated at the transmitting computer, col. 2, lines 51-59 et seq.; see also disclosure of a lag-recovery mechanism whereby if a lag is detected in the data transmission/reception, steps are taken to increase the data transmission or reception rate, such as by adjusting the data compression level, col. 7, lines 35-49 and col. 11, lines 9-22 et seq.).

Ground 6b: Carmel and Hill

Claims 1 and 4 are obvious under 35 U.S.C. § 103(a) in view of **Carmel** and **Hill**.

Regarding claim 1, **Carmel** teaches **a media player** (see disclosure that clients typically need only add a Java applet or plug-in to their existing web browser in order to receive and play back a broadcast, col. 7, lines 12-17 et seq.) for receiving **an audio or video program** as claimed, **the program comprising media data elements** (see disclosure that the data stream is divided into a sequence of segments or slices, preferably time slices, col. 2, lines 4-5 et seq.), from **a media source over an Internet protocol network** (see disclosure that a transmitting computer generates a data stream and broadcasts the data stream via a network server to a plurality of clients, and that the clients download the data stream from the server, preferably using an Internet protocol, col. 2, lines 1-15 et seq.; see also disclosure of broadcasting video data over the Internet, col. 1, lines 16-22 et seq.), and **playing the program for a user of the media player** (see disclosure that the one or more client computers decode the sequence and play back the data stream responsive to the indices thereof, col. 5, lines 22-25 et seq.), wherein **each of the media data elements is associated with a serial number** (see

disclosure that each slice of the data stream is assigned a respective slice index, col. 2, lines 4-7 and Fig. 3A:



),

comprising:

a) **a processor** (see disclosure that clients comprise conventional personal computers or workstations, col. 6, lines 38-40 et seq.; the Office notes that personal computers and workstations inherently include processors);

b) **a memory** (see disclosure that clients comprise conventional personal computers or workstations, col. 6, lines 38-40 et seq.; the Office notes that personal computers and workstations inherently include memory);

c) **a connection to the network** (see disclosure that a transmitting computer generates a data stream and broadcasts the data stream via a network server to a plurality of clients, and that the clients download the data stream from the server, preferably using an Internet protocol, col. 2, lines 1-15 et seq.; see

also disclosure of broadcasting video data over the Internet, col. 1, lines 16-22 et seq.); and

d) **media player software** (see disclosure that clients typically need only add a Java applet or plug-in to their existing web browser in order to receive and play back a broadcast, col. 7, lines 12-17 et seq.) comprising:

i) instructions to cause the media player to **request from the media source a predetermined number of media data elements** (see disclosure that the 'Fetch segment' module fetches the first segment and stores it in BUFF_A by making a request for URL(1), URL(2), and so on, until URL(A1), followed by execution of the 'Fetch next segment-and display present-segment' module which fetches the second segment and stores it in BUFF_B while simultaneously a client typically opens one or two HTTP links over which files 42, 44, 46, etc. are downloaded in successive alternation, col. 10, lines 36-40 et seq.);

ii) instructions to cause the media player to **receive media data elements sent to the media player by the media source and store the media data elements in the memory** (see disclosure that client computer 30 downloads and decodes files 42, 44, 46, etc., and reconstructs the multimedia data for output to the user through the use of timestamps,

col. 10, lines 35-54 et seq.; the Office notes that decoding and reconstructing/synchronizing the downloaded data would require said downloaded data first be stored in a memory);

iii) instructions to **implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer** (see disclosure that the operation of client 30 is controlled by a Java applet that includes facilities for carrying out the steps shown in Fig. 6A, col. 10, lines 27-34 et seq.; see also disclosure that the client selects an appropriate starting slice of data stream 40 and begins to download and decompress files 42, 44, 46, etc. in successive alternation and that client 30 reconstructs and outputs the multimedia data, col. 10, lines 35-50 et seq.; see also disclosure that the division of the data stream into slices and the inclusion of the slice indices in the data stream is used by the clients to maintain synchronization, col. 2, lines 17-21 et seq.);

iv) instructions to cause the media player to **play media data elements sequentially from the player buffer** (see disclosure that the one or

more client computers decode the sequence and play back the data stream responsive to the indices thereof, col. 5, lines 22-25 et seq.); and

v) instructions to cause the media player to **transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received** (see disclosure that the client selects an appropriate starting slice of data stream 40 and begins to download and decompress files 42, 44, 46, etc. in successive alternation and that client 30 reconstructs and outputs the multimedia data, col. 10, lines 35-50 et seq.; see also disclosure that client 30 monitors the time codes as the files are received in order to ensure that the transmission or reception is 'keeping up' with the input of data to the computer, col. 7, lines 36-40 et seq.).

To the extent that one could argue that **Carmel** fails to explicitly teach instructions to cause the media player to **transmit to the media source a request to**

send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received, **Hill** also teaches this limitation.

Specifically, **Hill** teaches that with respect to a high performance player for distributed, time-based media, the media player software comprises instructions to cause the media player to **transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received** (see disclosure that the buffer manager requests two new frames when the fill level is two frames less than the request threshold, one frame when the fill level is one frame less than the request threshold, and no frames otherwise, col. 9, lines 40-44 et seq.; see also disclosure that the request threshold represents the optimum number of future frames stored in the buffer at any given time, and is determined based on the characteristics of the particular system (e.g., I/O bandwidth, source latency, etc.), col. 6, lines 58-67 et seq.; see also

disclosure that the buffer manager uses the global frame number last requested as the starting point for determining which frame to request next, col. 10, lines 4-6 et seq.).

A POSITA would have found it obvious to combine the teachings of **Carmel** and **Hill** because the combination is merely a matter of applying a known technique to a known device for improvement to yield predictable results.

Carmel and **Hill** both teach video distribution systems over the Internet, including requesting data by serial number and to repeat such requests. Further, both maintain a record of the last serial number received. A POSITA under KSR at the time of the purported invention was made would have been motivated to combine the similar systems. For instance, a POSITA would have been motivated to implement **Hill's** threshold of buffering (**Hill**, at 6:58-67, 7:1-2, 9:40-44, 10:55-60, claims 8-9) to further enhance **Carmel's** real-time or synchronized playback. **Carmel**, at 2:1-21, 7:36-40, 10:50-54.

A POSITA would have been further motivated to implement such function and would have had a reasonable expectation of success in doing so considering **Carmel's** existing functionality. **Carmel** already teaches decoding, buffering, and playing sequential media slices (**Carmel** at 7:4- 17,7:59-8:5, 10:25-48, claim 10, Fig. 6A) as well as maintaining a record of the serial number of the last data element received (**Carmel** at

2:1-21, 8:18-26, 8:32-36, 10:25-48, Figs. 3, 6A). Thus, **Carmel** could implement **Hill's** buffer threshold seamlessly.

Accordingly, a POSITA would have been motivated to combine **Carmel** and **Hill**.

Regarding claim 4, **Carmel** additionally teaches the media player of claim 1, wherein the instructions for causing the media player to request from the media source a predetermined number of media data elements further causes the media player to **receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player** (see disclosure that with respect to a system for real-time multimedia broadcasting over a network, during normal operation the client computer monitors the downloading of data from the server in order to determine the amount of time required to convey each media slice and to verify that the slices are conveyed at a sufficient rate, and that when the data stream comprises multimedia data, the data rate should be generally equal to or faster than the rate at which the data are generated at the transmitting computer, col. 2, lines 51-59 et seq.; see also disclosure of a lag-recovery mechanism whereby if a lag is detected in the data transmission/reception, steps are taken to increase the data

transmission or reception rate, such as by adjusting the data compression level, col. 7, lines 35-49 and col. 11, lines 9-22 et seq.).

Ground 9a: Imai

Claim 1 is anticipated under 35 U.S.C. § 102(e) in view of **Imai**.

Regarding claim 1, **Imai** teaches **a media player** (see disclosure of file display program 130, col. 6, line 64 through col. 7, line 16 et seq., and Fig. 1; see also disclosure of viewer program 400, col. 10, line 47 through col. 12, line 20 et seq. and Figs. 6 and 7) for receiving **an audio or video program** as claimed, **the program comprising media data elements** (see disclosure that the term 'file' is used as a generic term for all the information which is formed by media such as text, figures, static images, video, audio, etc., and which can be electronically provided, and that one file may be formed by multiple media, col. 1, lines 28-31 et seq.), from **a media source over an Internet protocol network** (see disclosure that the information communication system is a client-server system for providing various files (hypertext in HTML format, static images, video images, audio data, etc.) from a server to a client by utilizing the network environment such as that of the Internet, for example, col. 20, lines 35-43 et seq.), and

playing the program for a user of the media player (see disclosure that the user or program requests the transfer and the display of the file by issuing the file transfer request, col. 22, lines 11-21 et seq.; see also step S511, at which the file requested by the user is displayed, col. 12, lines 1-2 and Fig. 7), wherein **each of the media data elements is associated with a serial number** (see disclosure of the use of file identifiers, such as file-A, which is a file identified by such an identifier A, and also the use of a URL which is an identifier for uniquely identifying a file, col. 22, lines 11-21 et seq.; see also disclosure of a file list composed of identifiers of the related files of this file-A, col. 22, lines 39-43 et seq.),

comprising:

a) **a processor** (see disclosure that the invention is implemented using conventional general purpose computers programmed according to the teachings of the specification, col. 33, lines 17-25 et seq.; the Office notes that conventional general purpose computers inherently include a processor);

b) **a memory** (see disclosure that file storing unit 134 is a program for caching (temporarily storing) a file received from the file server 110 into a cache region 121a of a storage medium 121, col. 21, lines 5-7 et seq. and Fig. 23);

c) **a connection to the network** (see disclosure that the information communication system is a client-server system for providing various files

(hypertext in HTML format, static images, video images, audio data, etc.) from a server to a client by utilizing the network environment such as that of the Internet, for example, col. 20, lines 35-43 et seq.); and

d) **media player software** (see disclosure of file display program 130, col. 6, line 64 through col. 7, line 16 et seq., and Fig. 1; see also disclosure of viewer program 400, col. 10, line 47 through col. 12, line 20 et seq. and Figs. 6 and 7) comprising:

i) instructions to cause the media player to **request from the media source a predetermined number of media data elements** (see disclosure that with respect to Fig. 25, <LIST> is a keyword indicating that this file is a file list, and the following character string indicates an identifier of the file-A which is requested by the user; the second and subsequent lines indicate the identifiers of the related files of this file-A, and also that a file having a name ending with '.au' is an audio file, col. 22, lines 39-47 et seq. and Fig. 25; see also disclosure that at steps S17 to S21, the multiple files request transmission processing to take out the identifier of the file from the produced file list one by one and sequentially issue the file request for each related file is started, col. 22, lines 55-59, col. 23, lines 28-45 et seq. and Fig. 24);

ii) instructions to cause the media player to **receive media data elements sent to the media player by the media source and store the media data elements in the memory** (see disclosure that file storing unit 134 is a program for caching (temporarily storing) a file received from the file server 110 into a cache region 121a of a storage medium 121, and that a stored position and a file name of a file cached into the cache region 121a are independently managed by the file display program 130, col. 21, lines 5-13 and col. 11, lines 5-17 et seq. and Fig. 23);

iii) instructions to **implement a player buffer manager, for managing a player buffer established in the memory, operable to maintain a record of the serial number of the last media data element that has been received and stored in the player buffer** (see disclosure that on the file requesting client 120, the file display program 130 is operating, and functions as the WWW client program (WWW browser), and includes a connection unit 131, a file request unit 132, a file receiving unit 133, a file storing unit 134, a file display unit 135, a request handling unit 136, a disconnection unit 137, and a multiple file transfer request unit 138, col. 20, lines 58-64 et seq.; see also disclosure that file storing unit 134 is a program for caching (temporarily storing) a file

received from the file server 110 into a cache region 121a of a storage medium 121, and that a stored position and a file name of a file cached into the cache region 121a are independently managed by the file display program 130, col. 21, lines 5-13 and col. 11, lines 5-17 et seq. and Fig. 23; see also disclosure that the file cached into cache region 121a is automatically allocated with a name and stored without a command from the user, unlike a usual file which is stored using a file name explicitly specified by the user, col. 21, lines 5-11 et seq.; see also disclosure that the multiple files transfer request unit 138 selects one not yet transferred file from the files listed in the file list, and gives its identifier to the file request unit 132 to command the issuance of the file transfer request (step S18, and that the file server program 150 then transfers the file specified by the file transfer request from the file request unit 132 to the file requesting client 120, where the file is received by the file receiving unit 133, and stored in the cache region 121a by the file storing unit 134 (step S21), col. 23, lines 29-41 et seq.; see also Figs. 23 and 24);

iv) instructions to cause the media player to **play media data elements sequentially from the player buffer** (see disclosure that a file is

requested from the file request unit 402, and the file requested by the file request unit 402 is received at the file receiving unit 403 and the received file is stored into the cache region 121a of the storage medium 121 by the file storing unit 404, that the file corresponding to the user's request is displayed on a viewer by the file display unit 405, and that the operations of the file request unit 402, the file receiving unit 403, and the file storing unit 404 according to the file list are repeated as many times as a number of files listed in the file list, col. 11, lines 5-17 et seq.; see also disclosure that the file display unit 135 is a program for displaying the requested file on a screen, col. 21, lines 25-29 et seq.); and

v) instructions to cause the media player to **transmit to the media source a request to send one or more media data elements, each identified by a serial number, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received** (see disclosure that the multiple files transfer request unit 138 selects one not yet transferred file from the files listed in the file list,

and gives its identifier to the file request unit 132 to command the issuance of the file transfer request (step S18, and that the file server program 150 then transfers the file specified by the file transfer request from the file request unit 132 to the file requesting client 120, where the file is received by the file receiving unit 133, and stored in the cache region 121a by the file storing unit 134 (step S21), and that the processing of steps S17 to S21 is repeatedly executed until there is no more not yet transferred file among the files listed in the file list, col. 23, lines 29-50 et seq.; see also Figs. 23 and 24).

Ground 9b: Imai and Carmel

Claim 4 is obvious under 35 U.S.C. § 103(a) in view of **Imai** and **Carmel**.

Regarding claim 4, **Imai** teaches the media player of claim 1 (see rejection of claim 1, Ground 9a, above).

Imai fails to explicitly teach **receipt of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player**. However, **Carmel** teaches this limitation[3].

Specifically, **Carmel** teaches that with respect to a system for real-time multimedia broadcasting over a network, during normal operation the client computer monitors the downloading of data from the server in order to determine the amount of time required to convey each media slice and to verify that the slices are conveyed at a sufficient rate. When the data stream comprises multimedia data, the data rate should be generally equal to or faster than the rate at which the data are generated at the transmitting computer. See col. 2, lines 51-59.

Carmel also discloses a lag-recovery mechanism whereby if a lag is detected in the data transmission/reception, steps are taken to increase the data transmission or reception rate, such as by adjusting the data compression level. See col. 7, lines 35-49 and col. 11, lines 9-22.

[3] This conclusion is consistent with the Final Written Decision on Remand by the PTAB in IPR2016-01238, 16 July 2020.

A POSITA would have found it obvious to combine the teachings of **Carmel** and **Imai** because the combination is merely a matter of applying a known technique to a known device for improvement to yield predictable results.

Imai and **Carmel** disclose similar client-pull systems, including requesting data by serial number using a file list or index file. A POSITA under KSR at the time of the purported invention was made would have been motivated to combine the teachings of **Imai** and **Carmel** given the similarity of the **Imai** and **Carmel** and a POSITA would have been motivated to review other client-pull systems to further efficiency and enhance **Imai's** functionality.

A POSITA would have been further motivated to implement **Carmel's** sending faster than playback rate and would have had a reasonable expectation of success in doing so considering **Imai's** existing server and client functionality. **Imai** already describes a cache region used to protect against network bandwidth changes and could store the data transmitted faster than the playback rate. **Imai**, at 21:5-18 ("This cache region 121a is utilized in order to relax the network congestion, and when a display request for the file stored in the cache region 121a is issued from the user, basically, the access to the file server 110 is not made and the corresponding file is read out from the cache region 121a and displayed on a screen"). Thus, **Imai** could implement **Carmel's** sending faster than playback rate seamlessly.

Accordingly, a POSITA would have been motivated to combine **Imai** and **Carmel**.

Ground 9c: Imai and Glaser

Claim 4 is obvious under 35 U.S.C. § 103(a) in view of **Imai** and **Glaser**.

Regarding claim 4, **Imai** teaches the media player of claim 1 (see rejection of claim 1, Ground 9a, above).

Imai fails to explicitly teach **receipt of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player**. However, **Glaser** also teaches this limitation.

Specifically, **Glaser** teaches that with respect to an audio-on-demand communication system, the client buffer receives data at a rate faster than the playback rate to maintain a predetermined threshold. For instance, at col. 22, lines 57-59, **Glaser** discloses that since the normal quality data blocks are transmitted at greater than real time, the buffer begins to refill and approach maximum capacity. At col. 21, lines 57-60, it is disclosed that when the subscriber PC determines that the buffers are near

maximum capacity (e.g., above 85% of capacity), this indicates that the normal quality data is being transferred in real time or greater than real time. At col. 20, lines 8-16, it is disclosed that the audio-on-demand system allows for greater than real time delivery of audio data to the subscriber PC in many cases.

A POSITA would have been motivated to combine **Imai** and **Glaser**. **Imai** and **Glaser** both teach video distribution systems.

A POSITA would have been motivated to implement **Glaser's** sending faster than the playback rate in **Imai** to enhance **Imai's** functionality.

A POSITA would have been further motivated to implement such function considering **Imai's** existing server structure and buffering ability to store the data sent faster than the playback rate. **Imai**, at Abstract, 10:15-25, 11:5-16, 21:32-43.

Accordingly, a POSITA would have been motivated to combine **Imai** and **Glaser**.

Conclusion

THIS ACTION IS MADE FINAL.

A shortened statutory period for response to this action is set to expire 2 months from the mailing date of this action.

Extensions of time under 37 CFR 1.136(a) do not apply in reexamination proceedings. The provisions of 37 CFR 1.136 apply only to "an applicant" and not to parties in a reexamination proceeding. Further, in 35 U.S.C. 305 and in 37 CFR 1.550(a), it is required that reexamination proceedings "will be conducted with special dispatch within the Office."

Extensions of time in reexamination proceedings are provided for in 37 CFR 1.550(c). A request for extension of time must specify the requested period of extension and it must be accompanied by the petition fee set forth in 37 CFR 1.17(g). Any request for an extension in a third party requested ex parte reexamination must be filed on or before the day on which action by the patent owner is due, and the mere filing of a request will not effect any extension of time. A request for an extension of time in a third party requested ex parte reexamination will be granted only for sufficient cause, and for a reasonable time specified. Any request for extension in a patent owner requested ex parte reexamination (including reexamination ordered under 35 U.S.C. 257) for up to two months from the time period set in the Office action must be filed no

later than two months from the expiration of the time period set in the Office action. A request for an extension in a patent owner requested ex parte reexamination for more than two months from the time period set in the Office action must be filed on or before the day on which action by the patent owner is due, and the mere filing of a request for an extension for more than two months will not effect the extension. The time for taking action in a patent owner requested ex parte reexamination will not be extended for more than two months from the time period set in the Office action in the absence of sufficient cause or for more than a reasonable time.

The filing of a timely first response to this final rejection will be construed as including a request to extend the shortened statutory period for an additional two months. In no event, however, will the statutory period for response expire later than SIX MONTHS from the mailing date of the final action. See MPEP § 2265.

The Patent Owner is reminded of the continuing responsibility under 37 CFR 1.565(a), to apprise the Office of any litigation activity, or other prior or concurrent proceeding, involving U.S. Patent No. 8,327,011 throughout the course of this reexamination proceeding. The third party requester is also reminded of the ability to similarly apprise the Office of any such activity or proceeding throughout the course of this reexamination proceeding. See MPEP §§ 2207, 2282 and 2286.

The Patent Owner is reminded that any proposed amendment to the specification and/or claims in the reexamination proceeding must comply with the provisions of 37 C.F.R. § 1.530(d)-(j), must be formally presented pursuant to 37 C.F.R. § 1.52(a) and (b), and must include any fees required by 37 C.F.R. § 1.20(c). See MPEP § 2250(IV) for examples to assist in the preparation of proper amendments in reexamination proceedings.

Submissions after the instant final Office action will be governed by the requirements of 37 CFR 1.116, after final rejection and 37 CFR 41.33 after appeal, which will be strictly enforced.

Any document filed by the Patent Owner must be served on the third party requester (or requesters in a merged proceeding) in the reexamination proceeding in the manner provided by 37 C.F.R. § 1.248. See 37 C.F.R. § 1.550(f) and MPEP § 2266.03.

All correspondence relating to this *ex parte* reexamination proceeding should be directed:

By EFS-Web:	Registered Users may submit correspondence via EFS-Web, at https://efs.uspto.gov/efile/myportal/efs-registered.
By Mail to:	Mail Stop *Ex Parte* Reexam Central Reexamination Unit Commissioner for Patents United States Patent & Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450
By FAX to:	(571) 273-9900 Central Reexamination Unit
By hand:	Customer Service Window Randolph Building 401 Dulany Street Alexandria, VA 22314

EFS-Web offers the benefit of quick submission to the particular area of the Office that needs to act on the correspondence. Also, EFS-Web submissions are "soft-scanned" (i.e., electronically uploaded) directly into the official file for the reexamination proceeding, which offers parties the opportunity to review the content of their submission after the "soft scanning" process is complete.

Any inquiry concerning this communication should be directed to the Central Reexamination Unit at telephone number 571-272-7705.

/LUKE S WASSUM/
Primary Examiner, Art Unit 3992

Conferees:

/ANGELA M LIE/
Primary Examiner, Art Unit 3992

/Michael Fuelling/
Supervisory Patent Examiner
Art Unit 3992

lsw
20 July 2022

PTO/AIA/31 (05-22)
Approved for use through 05/31/2024. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

NOTICE OF APPEAL FROM THE EXAMINER TO THE PATENT TRIAL AND APPEAL BOARD

Docket Number (Optional): 125737.538655

I hereby certify that this correspondence is being facsimile transmitted to the USPTO, transmitted via the USPTO's patent electronic filing system, or deposited with the United States Postal Service with sufficient postage in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on ______.	First Named Inventor Harold Edward Price	
	Application Number 90/014,833	Filed August 25, 2021
	For Streaming Media Delivery System	
Signature ______ Typed or printed name ______	Art Unit 3992	Examiner WASSUM, Luke S.

Applicant hereby **appeals** to the Patent Trial and Appeal Board from the last decision of the examiner.

The fee for this Notice of Appeal is (37 CFR 41.20(b)(1)) $ 840.00

☐ Applicant asserts small entity status. See 37 CFR 1.27. Therefore, the fee shown above is reduced by 50%, and the resulting fee is: $ ______

☐ Applicant certifies micro entity status. See 37 CFR 1.29. Therefore, the fee shown above is reduced by 75%, and the resulting fee is: $ ______
Form PTO/SB/15A or B or equivalent must either be enclosed or have been submitted previously.

☐ A check in the amount of the fee is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☐ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. ______.

☑ Payment made via USPTO's patent electronic filing system (Patent Center or EFS-Web).

☐ A petition for an extension of time under 37 CFR 1.136(a) (PTO/AIA/22 or equivalent) is enclosed.
For extensions of time in reexamination proceedings, see 37 CFR 1.550.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the

☐ applicant ☑ attorney or agent of record ☐ attorney or agent acting under 37 CFR 1.34
Registration number 50,478 Registration number ______

Signature /Mord M. Lewis/
Typed or printed name Mord Michael Lewis
Telephone Number 212-257-1639
Date September 28, 2022

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. Submit multiple forms if more than one signature is required, see below*.

☐ * Total of ______ forms are submitted.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. The United States Patent and Trademark Office (USPTO) collects the information in this record under authority of 35 U.S.C. 2. The USPTO's system of records is used to manage all applicant and owner information including name, citizenship, residence, post office address, and other information with respect to inventors and their legal representatives pertaining to the applicant's/owner's activities in connection with the invention for which a patent is sought or has been granted. The applicable Privacy Act System of Records Notice for the information collected in this form is COMMERCE/PAT-TM-7 Patent Application Files, available in the Federal Register at 78 FR 19243 (March 29, 2013). https://www.govinfo.gov/content/pkg/FR-2013-03-29/pdf/2013-07341.pdf

Routine uses of the information in this record may include disclosure to: 1) law enforcement, in the event that the system of records indicates a violation or potential violation of law; 2) a Federal, state, local, or international agency, in response to its request; 3) a contractor of the USPTO having need for the information in order to perform a contract; 4) the Department of Justice for determination of whether the Freedom of Information Act (FOIA) requires disclosure of the record; 5) a Member of Congress submitting a request involving an individual to whom the record pertains, when the individual has requested the Member's assistance with respect to the subject matter of the record; 6) a court, magistrate, or administrative tribunal, in the course of presenting evidence, including disclosures to opposing counsel in the course of settlement negotiations; 7) the Administrator, General Services Administration (GSA), or their designee, during an inspection of records conducted by GSA under authority of 44 U.S.C. 2904 and 2906, in accordance with the GSA regulations and any other relevant (i.e., GSA or Commerce) directive, where such disclosure shall not be used to make determinations about individuals; 8) another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)); 9) the Office of Personnel Management (OPM) for personnel research purposes; and 9) the Office of Management and Budget (OMB) for legislative coordination and clearance.

If you do not furnish the information requested on this form, the USPTO may not be able to process and/or examine your submission, which may result in termination of proceedings, abandonment of the application, and/or expiration of the patent.

Additional Uses

Additional USPTO uses of the information in this record may include disclosure to: 1) the International Bureau of the World Intellectual Property Organization, if the record is related to an international application filed under the Patent Cooperation Treaty; 2) the public i) after publication of the application pursuant to 35 U.S.C. 122(b), ii) after issuance of a patent pursuant to 35 U.S.C. 151, iii) if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections, or an issued patent, or iv) without publication of the application or patent under the specific circumstances provided for by 37 CFR 1.14(a)(1)(v)-(vii); and/or 3) the National Archives and Records Administration, for inspection of records.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventors: Harold Edward Price	Group Art Unit: 3992
Application No. 90/014,833	Examiner: WASSUM, Luke S.
Filed: August 25, 2021	Confirmation No. 1250
Title: STREAMING MEDIA DELIVERY SYSTEM	Patent No. 8,327,011

MAIL STOP APPEAL BRIEF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Appellant submits this Appeal Brief in response to the final office action mailed July 27, 2022 (the “FOA”), and further to the Notice of Appeal filed September 28, 2022. Appellant requests consideration of this appeal by the Patent Trial and Appeal Board.

[Ground 7c: *Withdrawn, not subject to appeal*;]

[Ground 8: *Withdrawn, not subject to appeal*;]

Ground 9a: Claim 1 is anticipated under 35 U.S.C. § 102(e) in view of U.S. Patent 5,987,510 to Imai *et al.* ("Imai");

Ground 9b: Claim 4 is obvious under 35 U.S.C. § 103(a) in view of Imai and Carmel; and

Ground 9c: Claim 4 is obvious under 35 U.S.C. § 103(a) in view of Imai and Glaser.

V. ARGUMENT

Points Common to Arguments

Limitation of Grounds to Patents and Printed Publications

Reexamination is limited to patents and printed publications. 37 CFR 1.552; MPEP 2258. The third-party Request, however, repeatedly makes assertions based on deposition testimony and the like, which are not patents or printed publications. Review procedures, including both IPRs and *ex parte* reexaminations, are both limited to patents and printed publications as bases for finding invalidity. It is important, in complying with this requirement, not to allow testimonial assessments, such as by depositions and declarations, to become "a replacement for documentary evidence for core factual findings in a determination of patentability." *K/S HIMPP v. Hear-Wear Technologies, LLC*, 751 F.3d 1362, 1365-66 (Fed. Cir. 2014). Such submissions are useful only in the narrow circumstances that enable an "instant and unquestionable demonstration as being well-known." *Id.* at 1366. Where limitations are not disclosed in the cited patents or printed publications, it would be improper to rely on expert declarations or deposition testimony as a substitute, which the Federal Circuit described, with reference to proceedings including reexamination in the CRU, as "a slippery slope

which would permit the examining process to deviate from the well-established and time-honored requirement that rejections be supported by evidence." *Id.*

Applicable *Phillips* claim construction standard where patent is expired

The '011 patent has expired and thus the proper claim construction standard is under *Phillips v. AWH* rather than the "broadest reasonable interpretation" ("BRI"). MPEP 2258(I)(G). Interpretations based on BRI must therefore be avoided because they lead to overly broad constructions and potentially to rejections that are not properly supported.

The FOA at 47 n.1 cites the PTAB decision on remand in IPR2016-0123 (previously submitted with the Request as Ex. 12), concerning different claims of a different patent, U.S. Patent No. 8,122,141 (Ex. 1 to the Request). Significantly, however, the Request omitted to submit the decision of the Federal Circuit on which the remand was based, in which the Federal Circuit had reversed the earlier decision of the PTAB. The Federal Circuit's decision, reported at 781 Fed. Appx. 1007 (Fed. Cir. 2019) was submitted as Patent Owner's Ex. 101.

The Federal Circuit decision contains important guidance on claim interpretation that is significant to the present reexamination, particularly its ruling that the relevant "rate" of data transfer under the claims at issue in that case was "the rate at which each requested media data element is transmitted from the server to the user computer." Ex. 101 at 10 (781 Fed. Appx. 1007, 1011 (Fed. Cir. 2019)). That ruling, plus the difference in wording between the claims at issue in that case and the claims here, are significant factors that must be taken into account in interpreting the present claims and considering them in view of the prior art.

Abbreviations for certain missing limitations

To avoid excessive verbiage in the arguments that follow, while remaining precise, the following shorthand abbreviations and reference labels are provided to

stand for the full text of various claim limitations asserted herein to be missing from the disclosures of the cited references ("**MLs**"). References herein to specific MLs are intended to reference the entire text of the corresponding ML, as set forth in the following table. Instead of repeating the full claim language at length over and over again in the following discussion, the abbreviations set forth below will be used.

Ref.	Abbreviation	Full Text of Limitation
ML1-1	"**maintain a record of the serial number of the last media data element**"	[Claim 1] instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to **maintain a record of the serial number of the last media data element that has been received and stored in the player buffer**
ML1-2	**request by serial number to the media server**	[Claim 1] **transmit to the media source a request to send one or more media data elements, each identified by a serial number**
ML1-3	"**repeat transmitting/maintain the player buffer**"	[Claim 1] **to repeat transmitting the requests to the media source for**

this regard is not to transmit segments more rapidly than the playback rate, but to send less data per frame to keep up with the playback rate.

The FOA cites Shteyn for the ability to select a lower bandwidth encoding, "ensuring that the segments are downloaded at a rate that exceeds the rate at which they are played out" (FOA at 44-45), citing to 4:20-23 of Shteyn. But the referenced citation does not support the assertion in the FOA (which, in fairness to the Examiner, comes from the third-party Request at 38, which is not prior art, and is in error in that it overstates the disclosure that it cites). The quoted language is not from Shteyn, it is from the Reexamination Request. The actual disclosure at 4:20-23 *et seq.* of Shteyn only reflects adapting encoding to make it easier to keep up with playback (adapt the encoding so as to send less data using the same bandwidth), not to transmit **more rapidly than** the playback rate, and thus does not support **ML4-1**.

Because of the foregoing deficiencies, Shteyn cannot be found to anticipate either claim 1 or claim 4.

B. GROUND 2A - ASSERTED OBVIOUSNESS OF CLAIM 4 OVER SHTEYN + CARMEL

The FOA would add Carmel (US 6,389,473 (Ex. 3 to the Request)), to meet **ML4-1**, and form an obviousness rejection of claim 4. *See* FOA at 45.

1. *Shteyn and Carmel, individually or in combination, fail to disclose ML4-1*

As noted in above, Shteyn's disclosures concerning adaptively changing the segment to be requested according to its bandwidth fails to address the rate at which the segment is transmitted or received, so as to send each requested segment faster than the playback rate (**ML4-1**). The FOA cites Carmel to overcome this deficiency. However, the FOA's basis for citing Carmel is the same adaptive bandwidth technique (changing the encoding to use available bandwidth) that is

insufficient to meet **ML4-1** under Shteyn. It is cumulative in this regard, and ineffective.

The FOA at 48 also cites Carmel's disclosure of addressing "lag" detected by "adjusting the data compression level." Carmel, 7:35-49 and 11:9-22. But this adds nothing to Shteyn, which as noted above, teaches the same thing, which likewise fails to meet **ML4-1**.

Thus, adding the disclosure of Carmel to that of Shteyn is of no help in meeting the "**receive elements more rapidly than the playback rate**" limitation **ML4-1**.

The FOA at 47-48 also cites Carmel's disclosure that "the data rate should be generally equal to or faster than the rate at which the data are generated at the transmitting computer" (Carmel, 2:51-59; FOA at 48). However, the cited teaching, that "the data rate should be generally equal to or faster than the [playback rate]," only goes to the **overall rate of transfer**. The Federal Circuit (*see* Ex. 101 herewith at 5), addressing a similar limitation in claim 10 of the related 8,122,141 patent (Ex. 1 to the Request), ruled that "the "rate" in claim 10 refers to the rate at which **each** requested media data element is transmitted from the server to the user computer" and that "[t]he rate limitation in claim 10 therefore refers to the rate at which requested media data elements are sent, not the overall rate at which data is transmitted from the server to the user computer." Ex. 101 at 10. Thus, the limitation in question was held to be directed to the **requested elements**, and specifically to "**each**" requested element, as reflected in the Federal Circuit decision. Claim 4 recites that player instructions cause "the predetermined number of media data elements," meaning each of them, to be received at a rate more rapid than the playback rate.

Moreover, as addressed above with regard to Shteyn, the "predetermined number of media data elements" of **ML4-1** refers back to the "predetermined

number of media data elements" requested per claim 1, which unquestionably includes the first element requested, *i.e.*, on streaming startup. The antecedent "**instructions to cause the media player to request from the media source a predetermined number of media data elements**" recited in claim 1 necessarily implies that those "instructions" would do as recited for the first request. The "each requested element" per the Federal Circuit decision (Ex. 101) implies the same, as "each" would include the first. Likewise, although the claim language itself is sufficiently clear on this point (with the initially recited "request" being followed by "repeating" such requests to the end of the program stream), the specification reinforces this understanding, where at the outset it notes a prime object to avoid "[d]elayed starts" ('011 patent, 4:35-36), stating in the Summary of the Invention, that in doing so, "Once a connection is made … First, media data is sent to the user at the highest rate that the data connection between the server and the user computer will support" ('011 patent, 5:46-48), and with regard to the particular embodiment relevant to this claim, that: "The media data will be transmitted to the user computer as fast as the data connection between the user computer and the server will allow. '011 patent, 8:42-44.

Carmel's disclosure of "generally" sending at a rate "equal to or faster than" the playback rate certainly does not constitute a teaching to send the first requested element of the stream faster than the playback rate. Nor do any other of the cited disclosures of Carmel (for example, the disclosure of switching to a different encoding), which address events **after** the start of streaming, to adjust for detected lag. Moreover, the very existence of a "lag" as disclosed in Carmel means that individual elements of the stream have been transmitted a rate that is *lower* than the playback rate.

In a footnote, the FOA also cites a PTAB decision on the '141 patent on remand from the Federal Circuit. FOA at 47 n.1. The PTAB decision is not binding

playback rate, and no rationale is provided for how the combination could do so. Neither reference discloses this actual claimed feature, and no rationale is articulated for combining them so as to achieve a further aspect that neither of the references teaches.

D. GROUND 3A - ASSERTED ANTICIPATION OF CLAIMS 1 AND 4 BY HILL

1. ML1-2: … transmit to the media source a request to send one or more media data elements, each identified by a serial number

Hill (US 6,005,600, Ex. 10 to the Request) concerns a system for viewing a moving picture using a workstation, over a network. Hill discloses a workstation (player) having a buffer for video frames to be played back, and mechanisms for requesting specific frames to make up for the player's buffer falling more than one frame behind during streaming. However, these operations are disclosed as being performed primarily **within** the player, by a "buffer manager" (118), through "source manager" (120)—both of those components being local to the workstation, 104. *See* Hill, Fig. 1 for the component layout. Specifically, it is the (workstation-internal) "source manager" 120, rather than (external) "source" 122, that transfers the requested frame to the buffer manager 120. *See* Hill, 6:30-31.

Claims 1 and 4 concern requests from a media player to a "media source," not requests to player-side components such as the "buffer manager" or "source manager" of Hill, which are both internal to the player.

Hill does disclose a media source external to the player. *See* source 122 in Fig. 1 of Hill. However, while describing the internal operation of the "source manager" 120 relative to "buffer manager "118" at considerable length, Hill does not disclose how source manager 120 (within the player) actually obtains media data from source 122 (outside the player). In a "black box" manner, relative to the actual media source 122, the disclosure states: "The source manager 120 provides

the necessary interface (*i.e.*, hardware or software) for communicating with a source 122." Hill, 4:28-30. "The source manager 120 in step 316 retrieves the requested frame data from the source 122." Hill, 6:14-16. But there is no disclosure of how media data elements are requested from source 122.

Thus, Hill fails to disclose a **request by serial number to the media server (ML1-2)**, because there is no description of the request to source 120 in Hill that meets **ML1-2**. Hill discloses that there is a request of some type to the media server, but not what the request specifies.

2. ML1-1: ... maintain a record of the serial number of the last media data element that has been received

Hill also fails to disclose "**maintain a record of the serial number of the last media data element that has been received**" (**ML1-1**). Hill discloses individual frame requests in the particular circumstance where it needs catch up the buffer if it is falling behind in streaming. Hill, 7:11-16. However, it refers to these particular frames as simply one or two "new" frames. To determine which frame to request in that limited circumstance, Hill discloses "adding one to the last **request**." Hill, 7:26-28, There is no teaching in Hill to maintain a record of the serial number of the last **received** segment.

3. Claim 4, ML4-1: ... wherein the instructions ... further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player

With regard to claim 4, Hill fails to disclose **ML4-1**, to "**receive elements more rapidly than [the playback rate]**." Hill discloses requesting one (1) new frame when the player buffer level falls below a specified level. Hill, 7:12-16. In the particular case in which the buffer should fall more than one frame behind, the buffer manager will request two new frames. Hill, 7:15-17. The disclosure notes

that a request for two frames is the preferred limit, "in order to avoid overloading sources." Hill, 7:17-20. While the limit of two is not necessarily the maximum (Hill, 7:22-24), there is no teaching or suggestion in Hill that the same mechanism could be used to request every segment in the program, and a fair reading of the reference is that it suggests that, while two segments may not be the upper limit, requesting the entire stream via this mechanism would put undue loads on the streaming source.

Hill indeed discloses that it "assume[s] that the sources 122 will always be able to provide frames at the normal play rate." Hill, 8:21-22. It does not teach sending all requested elements more rapidly than the playback rate. In this regard, the disclosure, in referencing the requests for one frame, or two frames where the buffer has fallen more than one frame behind, also reflects the corresponding data transfer rates: "Hence, if the play rate is 24 frames per second, the sources will not be asked to supply more than 48 frames per second." Hill, 7:20-22. It is therefore apparent that in the case the buffer is only one frame behind, the request will be for one frame encoded at 24 frames per second. If the request is for two frames, the server will have to supply 48 frames per second—only the latter necessarily being faster than the playback rate. However, there is no teaching that the frames requested one frame at a time should be sent or received any faster than the 24-frame-per-second playback rate. This does not meet the limitation wherein each requested data element is received more rapidly than the playback rate.

Furthermore, Hill fails to teach sending the first requested element faster than the playback rate (see related discussions above with regard to this same issues concerning Shteyn and Carmel, and the fact that **ML4-1** is referring to the initial requested predetermined number of elements). The cited disclosures from Hill (FOA at 59) concern where the streaming has already fallen one or more frames behind. As noted above, these disclosures do not address the

transmission/reception rate for "each requested element," or the "first" element (which follows from "each").

Hence, Hill fails to teach **ML4-1**, to "receive elements more rapidly than the playback rate."

E. GROUND 3B - ASSERTED OBVIOUSNESS OF CLAIM 4 OVER HILL + CARMEL

The FOA at 59 cites Carmel for **ML4-1** (**receive elements more rapidly than the playback rate),** to the extent missing from Hill (which it is).

The FOA's citation of Carmel to supplement Hill's failure to disclose **ML4-1** raises the same issues that were presented where the FOA likewise relied on Carmel to supplement Shteyn's failure to disclose **ML4-1**. As already discussed at length, with regard to the same issue concerning a § 103 combination with Shteyn, Carmel does not teach **ML4-1**. Rather than repeat here again at full length the same points already set forth with regard to Carmel, Patent Owner refers to the detailed discussion of Carmel relative to **ML4-1**, which may be found at pages **25** through **30** in this paper, above. Furthermore, Hill itself (Hill, 7:17-20) appears to **teach away** from any approach that would require sending the first-requested "predetermined number" of elements of program stream more rapidly than the playback rate, giving as an example requesting only two frames "to avoid overloading the sources," clearly suggesting that any sustained transmission above the playback rate should be avoided.

Moreover, as also noted, in addition to not teaching **ML4-1** (receive the predetermined number of elements more rapidly than the playback rate), Carmel does not teach other limitations missing from Hill, such as **ML1-1**, maintain a record of the serial number of the last media data element that has been received.

Merely adding Carmel to form a § 103 rejection does not cure the deficiencies of Hill and does not address at least three limitations of the claims.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/014,833	08/25/2021	8327011	125737.538655	1250

7590 03/15/2023

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES, LLC
231 SOMERVILLE ROAD
BEDMINSTER, NJ 07921

EXAMINER	
WASSUM, LUKE S	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
03/15/2023	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

VENABLE LLP
1290 AVENUE OF THE AMERICAS
NEW YORK, NY 10104-3800

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. *90/014,833* .

PATENT UNDER REEXAMINATION *8327011* .

ART UNIT *3992* .

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

PTOL-465 (Rev.07-04)

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Application Number: 90/014,833
Filing Date: 25 Aug 2021
Appellant(s): WAG Acquisition, LLC

____
M. Michael Lewis
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 28 November 2022.

(1) Grounds of Rejection to be Reviewed on Appeal

Every ground of rejection set forth in the Office action dated 27 July 2022 from which the appeal is taken is being maintained by the examiner except for the grounds of rejection (if any) listed under the subheading "WITHDRAWN REJECTIONS." New grounds of rejection (if any) are provided under the subheading "NEW GROUNDS OF REJECTION."

(2) Response to Argument

<u>Preliminary Discussion</u>

A significant portion of Patent Owner's arguments concern the claim feature that each of the media data elements are associated with a serial number, and that the serial number identifies the media data element in a request. A brief discussion of the correct construction of the claimed 'serial number' may be helpful.

The '011 patent has been the subject of prior litigation[1][2].

---

[1] *WAG Acquisition, LLC v. Gattyán Group S.à r.l., et al.*, 2:14cv02832.
[2] *WAG Acquisition, LLC v. Flying Crocodile, Inc., d/b/a FCI, Inc.*, et al., 2:19cv01278.

During this litigation, Patent Owner submitted a number of claim construction briefs, arguing that the term 'serial number' should be accorded a broad interpretation.

For instance, in Plaintiff's Opening Claim Construction Brief filed in the *WAG Acquisition, LLC v. Gattyán Group S.à r.l., et al.* litigation (Requester's Exhibit 15), Patent Owner contended that under the *Phillips* claim construction standard "[n]othing in the plain meaning of "serial number" requires that it be "consecutive"", and that "nothing in the common understanding of "serial number" requires that it even be made wholly and exclusively from numbers." "A POSITA would recognize that this term simply means an identifier that corresponds to a sequence in connection with the media data elements." (pages 22-23.)

These arguments are reiterated in Plaintiff's Responsive Claim Construction Brief (Requester's Exhibit 16), filed 4 March 2021, at page 19.

In the *WAG Acquisition, LLC v. Flying Crocodile, Inc., d/b/a FCI, Inc*. litigation, Patent Owner made similar arguments in the Plaintiff's Opening Claim Construction Brief (Requester's Exhibit 17), filed 24 March 2021, at pages 19-20.

In the Plaintiff's Responsive Claim Construction Brief (Requester's Exhibit 18), filed 14 April 2021, Patent Owner contended at pages 9-10 that "[j]ust because there is a

"series" of serial numbers/identifiers does not mean that they must be consecutive. Non-consecutive numbers can represent a series."

> Moreover, FCI tries to conflate the fact that the media data elements themselves are "sequential" with some requirements that the identifiers themselves are consecutive. The series of the identifiers / numbers *corresponds* to a sequence. That sequence is the multimedia data itself. All multimedia data is in a time sequence. The identifiers / numbers are in a series that identifies that sequence, but the identifiers / numbers are not themselves necessarily in a sequence or consecutive, they are merely in a defined series so that they correspond to the sequence of the multimedia data elements. They could even be random.

In light of the construction offered by Patent Owner in the above-cited claim construction briefs, the examiner has adopted Patent Owner's proposed broad construction for the term 'serial number'.

<u>Section V.A.1</u>

On pages 15-20 of the Brief, Patent Owner argues that **Shteyn** fails to disclose the claimed <u>transmission of a request to the media source to send one or more data elements, each *identified by a serial number*</u>.

The examiner respectfully disagrees.

As discussed above, the claimed 'serial number' is given a broad interpretation, consistent with Patent Owner's proposed construction in past litigation. As such, the claimed serial numbers need not be numeric, and need not be sequential, but are only required to be identifiers corresponding to a specific media data element.

Shteyn discloses media [content] files that are split into multiple parts [segments]. See col. 1, lines 65-66.

Also disclosed is that 'information-describing content' is maintained for the media [content] files, coded in XML. Specifically, the XML coding includes information regarding the media file overall (e.g., title, artist), as well as information regarding each of the individual segments (analogous to the claimed media data elements) (e.g., segment label, length, format, location, minimum bandwidth).

A sample of an XML coding is illustrated in Fig. 2:

```
<XML>
    <title>
    The best ever music
    </title>
    <artist>
    V.R. Famous
    </artist>
    <parts>                 (Preferred format)
        <part1>
            <length> 1024 </length>
            <format> MP3 </format>
            <location> ftp://137.27.52.87 </location>
            <min_bandwidth> 10,000 </min_bandwidth>
        </part1>

                            (Alternative format)
        </part1_alt>
            <length> 512 </length>
            <format> OTHER </format>
            <location> http:// yevgeniynet/ .... _</location>
            <min_bandwidth> 8,000 </min_bandwidth>
        </part_alt1>

........

    </parts>
</XML>
```

Given the broad interpretation of the claimed 'serial number' discussed above, either the segment label (e.g., part1, part1_alt) or the location could serve to identify a specific segment, and so would satisfy the preamble limitation that <u>each of the media elements is associated with a serial number</u>.

As an initial matter, the examiner notes that **Shteyn** alone teaches this claim feature, for the reasons presented above with respect to Patent Owner arguments of Section V.A.3.

Carmel also teaches this limitation.

Patent Owner argues that **Carmel's** disclosure that "the data rate should be generally equal or faster than the [playback rate]" only goes to the **overall rate of transfer**, which is inconsistent with the construction applied to the term "rate" by the Federal Circuit's decision in IPR2016-01238 (Patent Owner's Exhibit 101), which required that the rate of **each** requested element is more rapid than the playback rate.

However, the PTAB, in its Final Written Decision on Remand (Requester's Exhibit 12) (responsive to the above-cited Federal Circuit decision) found that even under the Federal Circuit's construction of the term 'rate', **Carmel's** disclosure that <u>under normal operation, the data rate should be generally equal to or faster than the playback rate</u> anticipated the claimed sending of media data elements at a rate more rapid than [the playback rate]. See Final Written Decision, pages 18-21.

Therein, the PTAB found that the Patent Owner "relies on an improper importation of an additional limitation into the claim, namely that *all* requested media

data elements must be sent by the server at a rate more rapid than the rate at which the streaming media is played back by a user."

They further noted on page 19 that a claim limitation is met by a device that performs the function "some of the time."[3]

The PTAB provided further elaboration on pages 20-21:

> "For method claims, it is well established that "part-time" satisfaction of a method claim is sufficient to establish anticipation. See, e.g., *Hewlett-Packard Co. v. Mustek Svs. Inc.*, 340 F.3d 1314 Fed. Cir. 2003) ("a prior art product that sometimes, but not always, embodies a claimed method nonetheless teaches that aspect of the invention"). The same principle analogously applies to the software aspects of apparatus claim 10. That is, the mere fact that the server in Carmel might comprise a machine-readable, executable routine containing instructions to cause the server to send media data elements to the user system at a rate less rapid than the playback rate does not diminish the fact that it also contains instructions for sending media data elements at a rate more rapid than the playback rate."

In the case of the '011 patent, the claims are directed to instructions that cause the media player to <u>receive</u> the media data elements at a rate more rapid than the playback rate. The PTAB's line of reasoning with respect to the '141 patent also applies to the '011 patent, however. **Carmel** discloses the transmission of data at a rate generally

[3] *Broadcom Corporation v. Emulex Corporation*, 732 F.3d 1325, 1333 (Fed. Cir. 2013).

equal or faster than the [playback rate] during normal operation, meaning that **Carmel** discloses the claimed instructions that cause the media player to receive data elements at a rate more rapid than the playback rate, even given the Federal Circuit's construction of the term 'rate'.

For the same reasons, **Carmel's** disclosure of a 'lag recovery' operation also reads on the claimed limitation. With respect to this operation, **Carmel** discloses that '[i]n the event that a lag is detected, steps are taken to increase the data transmission reception rate', such as through a client selection of a lower quality level.

The examiner also notes once again that the claims are directed to instructions on the client device. The claimed instructions are required to <u>cause</u> data to be received at the media player at a rate faster than the playback rate. However, there is no way to write instructions that can *cause* data to be received at any specific rate. The rate at which data is received would be wholly dependent upon the rate at which it was transmitted, and by network conditions such as bandwidth and congestion.

Patent Owner argues that the PTAB decision on remand is inapplicable in this case for a number of reasons.

Furthermore, the combination of **Shteyn** and **Carmel** is predicated on the assumption that **Shteyn** fails to disclose the claimed reception of media data elements more rapidly than the replay rate. In that case, the motivation for incorporating **Carmel's** mechanism for doing this is evident; it would allow the reception of media data at a rate sufficient to prevent pauses in the playback while the system waits for additional media data to be downloaded. As is stated in the Office action, this would involve merely substituting one known mechanism for downloading media data for another.

Section V.C

On pages 32-34 of the Brief, Patent Owner argues that the combination of **Shteyn** and **Glaser** fails to disclose the claimed media player receives media data elements at a rate more rapid than the playback rate.

The examiner respectfully disagrees.

Specifically, Patent Owner again argues that the claim language, and specifically the term "rate", read in light of the Federal Circuit's decision in IPR2016-01238, requires

that all media data elements are required to be received at a rate more rapid than the playback rate.

Initially, the examiner again notes that that claim 4 is directed to software operating on the client media player, and specifically to the rate at which media data elements are received. However, one could not write software that would **cause** data to be received at any given rate (as is claimed), since reception rate is fully dependent upon the rate at which the data is transmitted, and further upon network conditions such as bandwidth and congestion.

Furthermore, as discussed previously with respect to Section V.B.1 above, the PTAB, in their Final Written Decision responsive to the cited Federal Circuit decision, noted that a disclosure that the reception rate exceeds the playback rate in *some* cases is evidence that the claimed instructions for receiving media data elements at a rate more rapid than the playback rate are necessarily present in the prior art.

Therefore, **Glaser's** disclosure at col. 22, lines 57-59 that "[a]s the normal quality data blocks are transmitted at greater than real time, the buffer **315** begins to refill and approach maximum capacity" reads on Patent Owner's claim 4.

It would have been obvious to incorporate such a feature into **Shteyn**, since it would allow the reception of media data at a rate sufficient to prevent pauses in the playback while the system waits for additional media data to be downloaded.

Section V.D.1

On pages 34-35 of the Brief, Patent Owner argues that **Hill** fails to disclose the claimed transmission of a request to the media source to send one or more data elements, each *identified by a serial number*.

The examiner respectfully disagrees.

Specifically, Patent Owner argues that **Hill** discloses a media source external to the media player, but fails to disclose how source manager 120 (within the player) actually obtains media data from source 122 (outside the player). "Hill discloses that there is a request of some type to the media server, but not what the request specifies."

However, **Hill** discloses that if new frames are required, the buffer manager uses the cliplist manager to retrieve those frames from the appropriate source (col. 5, lines 24-28). Also disclosed is the fact that when a request for new frames is issued to the

cliplist manager, it converts the global frame number to a clip number and a local frame number, and then passes the request to the appropriate source manager which retrieves the requested frame data from the corresponding source (col. 6, lines 7-16).

Notably, **Hill** discloses that while performing the retrieval, the source accesses the *requested* frame (col. 5, lines 29-30). Since a given source stores any number of frames, a request for a frame would necessarily include information to identify the requested frame (such as the local frame number that has been decoded by the cliplist manager from the global frame number in the request from the buffer manager).

See also exemplary disclosure that in the case where global frame 106 is required, cliplist manager maps global frame 106 to clip 2, local frame 17, *then requests this frame* from the second source manager (col. 12, lines 31-34).

Hill's local frame number (or alternately, the combination of clip number and local frame number), identifying the desired media frame, reads on the claimed serial number, identifying the desired media data element.

Section V.D.2

On page 35 of the Brief, Patent Owner argues that **Hill** fails to disclose the claimed maintenance of a record of the serial number of the last media data element that has been received and stored in the player buffer.

The examiner respectfully disagrees.

Hill discloses that the system keeps track of not only the last media data element requested, but also the last one received.

For instance, at col. 12, lines 27-40, **Hill** discloses that the buffer manager checks to see if global frames 96 through 106 are currently being stored, *or whether they have been requested*. This means that the system differentiates between requested frames and *received* frames, by maintaining records of not only which frame has been last requested, but which has been last received (and stored in the buffer). When the frame is received, the buffer manager inserts the received frame into the buffer as global frame 106 (and updates its record to reflect the insertion of global frame 106 into the buffer.)

<u>Section V.D.3</u>

On pages 35-37 of the Brief, Patent Owner argues that **Hill** fails to disclose the claimed <u>media player receives media data elements at a rate more rapid than the playback rate</u>.

The examiner respectfully disagrees.

Patent Owner's arguments largely mirror arguments presented with respect to Section V.A.3 above, where they were previously addressed.

As noted above, because the instant claims are directed to software on the client media player, and because the rate of reception of media data elements is completely dependent upon the transmission rate and network conditions, claim 4 cannot be properly interpreted as requiring the reception rate of each media data element to exceed the playback rate. At best, the software is required to be <u>capable of</u> receiving the media data elements at a rate higher than the playback rate.

Hill's specification at col. 7, lines 11-24, that the system can download multiple frames, such that if the playback rate is 24 frames/second, the download rate can be up to 48 frames/second, by itself discloses that the system is capable of receiving the frame data at a rate that exceeds the playback rate, as required by the language of claim 4.

Furthermore, as discussed previously with respect to Section V.B.1 above, the PTAB, in their Final Written Decision responsive to the cited Federal Circuit decision, noted that a disclosure that the reception rate exceeds the playback rate in *some* cases is evidence that the claimed instructions for receiving media data elements at a rate more rapid than the playback rate are necessarily present in the prior art.

Therefore, **Hill's** disclosure at col. 7, lines 11-24, that the system can download multiple frames, such that if the playback rate is 24 frames/second, the download rate can be up to 48 frames/second reads on Patent Owner's claim 4.

Patent Owner also argues that **Hill** fails to teach sending the *first* requested element faster than the playback rate.

However, it is noted that there is no specific claim language that differentiates the reception rate of the *first* media data element from other subsequently received media data elements. Since **Hill** discloses the reception of media data elements at a rate more rapid than the playback rate (for the reasons discussed above), the claim limitation is satisfied.

Section V.E

On pages 37-38 of the Brief, Patent Owner argues that the combination of **Hill** and **Carmel** fails to disclose the claimed media player receives media data elements at a rate more rapid than the playback rate.

The examiner respectfully disagrees.

As an initial matter, the examiner notes that **Hill** alone teaches this claim feature, for the reasons presented above with respect to Patent Owner arguments of Section V.D.3.

Carmel also teaches this limitation.

Patent Owner refers back to the arguments presented in Section V.B with respect to the rejections in view of **Shteyn** and **Carmel**. These arguments have been addressed above.

Patent Owner also argues that **Hill** teaches away from any approach that would require sending the first-requested "predetermined number" of elements of program stream more rapidly than the playback rate.

However, as discussed above with respect to Section V.D.3, there is no specific claim language that differentiates the reception rate of the *first* media data element from other subsequently received media data elements. Since **Hill** discloses the reception of media data elements at a rate more rapid than the playback rate (for the reasons discussed above), the claim limitation is satisfied.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventors: Harold Edward Price	Group Art Unit: 3992
Application No. 90/014,833	Examiner: WASSUM, Luke S.
Filed: August 25, 2021	Confirmation No. 1250
Title: STREAMING MEDIA DELIVERY SYSTEM	Patent No. 8,327,011

MAIL STOP APPEAL BRIEF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF

Appellant submits this Reply Brief in response to the Examiner's Answer dated March 15, 2023 (the "Answer").

relevant to **ML4-1**, are the ones sent *more slowly* than the playback rate. *See* Glaser, 17:36-40, 20:21-21-35. The underlying alleged motivation to combine as present in the Answer, is that "it would allow the reception of media data at a rate sufficient to prevent pauses in the playback while the system waits for additional media data to be downloaded" (Answer at 25). But Shteyn already has the same tactic in its own arsenal, which is to switch encoding levels. There is no motivation to attempt to adapt an incompatible mechanism from a second reference simply to do what the base reference already does.

D. GROUND 3A - ASSERTED ANTICIPATION OF CLAIMS 1 AND 4 BY HILL

1. ML1-2: … transmit to the media source a request to send one or more media data elements, each identified by a serial number

The entire premise in the Answer concerning alleged disclosure in Hill of **ML1-2**, is rooted in inherency, in which the Answer argues that "[s]ince a given source stores any number of frames, a request for a frame would necessarily include information to identify the requested frame (such as the local frame number that has been decoded by the cliplist manager from the global frame number in the request from the buffer manager)." Answer at 26. The Answer provides no factual underpinning as to why any undisclosed request in Hill must *necessarily include* a requested frame number. It is uncontested that Hill is silent

on the issue as to how requests for media are made, disclosing only that "source manager 120 in step 316 retrieves the requested frame data from the source 122" (Hill, 6:14-16), and nothing more. Given that inherency is a question of fact, without any factual support, the Answer's inherency argument must fail. *PAR Pharm.*, 773 F.3d at 1194 ("inherent teaching of a prior art reference is a question of fact."); *id.*, at 1195-96 (establishing inherency requires "meet[ing] a high standard" to establish that a "limitation at issue necessarily must be present" in the prior art). Just to scratch the surface of possibilities, Hill's client could request a download of the entire media file, or could implement a "request next" protocol like Shteyn. Merely selecting one alternative out of many does not rise to inherency.

2. ML1-1: … maintain a record of the serial number of the last media data element that has been received

The Answer relies upon disclosure at 12:27-40 in Hill as meeting limitation **ML1-1**, in which Hill discloses "buffer manager 118 will check to see if global frames 96 through 106 are currently being stored, or whether they have been requested," in which cliplist manager 116 requests a specific un-stored frame from source manager 120 and, when this "frame is received, the source manager 120 will arrange with the buffer manager 118 to have it inserted into the buffer as global frame 106." According to the Answer, "[t]his means that the system

differentiates between requested frames and received frames, by maintaining records of not only which frame has been last requested, but which has been last received (and stored in the buffer)." Answer at 27.

Although this disclosure indicates that Hill keeps track of what frames have been *stored*, it does so according to internal numbering. There is nothing in this disclosure that teaches that Hill necessarily keeps track of any identifier argued to be used by source manager 120 to request frames from source 122 of Hill, or even the internal numbering of the *last frame* that has been *received*, as required of **ML1-1**.

For at least the reasons stated so far, Hill fails to anticipate claim 1.

3. Claim 4, ML4-1: … wherein the instructions … further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player

As with allegations of disclosures of **ML4-1** in Carmel and Shteyn above, the Answer's logic is premised upon a faulty construction of **ML4-1**, arguing that "a disclosure that the reception rate exceeds the playback rate in *some* cases is evidence that the claimed instructions for receiving media data elements at a rate more rapid than the playback rate are necessarily present in the prior art." Answer at 28 (emphasis in original). But, as properly construed, **ML4-1** is not directed to

merely *some* of the requested media data elements but to "the predetermined number" of them and more particularly *each* of the predetermined number.

Although Hill discloses that, under certain circumstances, *some* frames may be sent faster than the playback rate, there is no disclosure in Hill that *every* requested frame is received faster than the playback rate, or that "the predetermined number" is received at such a rate, and the Answer points to no such disclosure.

For at least these reasons, Hill does not anticipate claim 4.

E. GROUND 3B - ASSERTED OBVIOUSNESS OF CLAIM 4 OVER HILL + CARMEL

1. Carmel fails to disclose ML4-1

As discussed at length above with respect to Ground 2A, Carmel does not disclose or suggest the features of **ML4-1**.

ML4-1 applies to "the predetermined number" of elements and more particularly *each* of the predetermined number, it applies with equal force to the first such requested element. The Answer's point that "there is no specific claim language that differentiates the reception rate of the *first* media data element from other subsequently received media data elements" is off point. Answer at 30 (emphasis in original). The claim language applies to the predetermined number of elements, which as noted above includes the first element.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
U.S. Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO./ CONTROL NO.	FILING DATE	FIRST NAMED INVENTOR / PATENT IN REEXAMINATION	ATTORNEY DOCKET NO.
90/014,836	08/25/2021	8364839	125737.538655

7590 10/05/2023

VENABLE LLP
1290 AVENUE OF THE AMERICAS
NEW YORK, NY 10104-3800

EXAMINER	
CAMPBELL, JOSHUA D	
ART UNIT	**PAPER NUMBER**
3992	
MAIL DATE	**DELIVERY MODE**
10/05/2023	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

MOD PTOL-90A (Rev.06/08)

RECORD OF ORAL HEARING

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte WAG ACQUISITION, LLC

Appeal 2023-002525 (Reexamination Control 90/014,833)
Appeal 2023-003028 (Reexamination Control 90/014,836)
Appeal 2023-003319 (Reexamination Control 90/014,834)
Technology Center 3900

Oral Hearing Held: September 18, 2023

Before JOHN A. JEFFERY, ERIC B. CHEN, and
BRIAN J. MCNAMARA, *Administrative Patent Judges*.

APPEARANCES:

ON BEHALF OF THE APPELLANTS:

RONALD ABRAMSON, ESQUIRE
Liston Abramson LLP
The Chrysler Building
405 Lexington Avenue, 46th Floor
New York, New York 10174

The above-entitled matter came on for hearing on Monday, September 18, 2023, commencing at 1:00 p.m., at the U.S. Patent and Trademark Office, 600 Dulany Street, Alexandria, Virginia.

PROCEEDINGS

\- - - - -

JUDGE CHEN: Okay, Mr. Abramson, welcome to the board. This is Judge Chen, and with me today are Judges Jeffery and McNamara. Before we get started, just a few administrative matters. Number one, if you're not speaking, please mute yourself. Number two -- number two, please identify yourself each time you speak. This helps the court reporter prepare an accurate transcript. Number three, we have the entire record before us, including demonstratives.

If you are identifying demonstrative by page number, please provide a few seconds for us to access that document. And lastly, please be aware that members of the public may be listening in on this oral hearing. That being said, please state your name and law form for the record.

MR. ABRAMSON: Ronald Abramson, Liston Abramson LLP.

JUDGE CHEN: Okay. Thank you, Mr. Abramson. You have about 20 minutes, and you may begin. And just initially for, excuse me. This is Appeal Number 2023-002525. Mr. Abramson, you may begin when you're ready. And we would ask that you -- we're particularly interested in the Hill reference with respect to this appeal. So, if you need a few minutes to organize your notes, we can provide that for you.

MR. ABRAMSON: Yeah, the notice I got was all three appeals combined, so we're doing them one by one. Which one are we — which one is this?

JUDGE CHEN: Okay, this is 2023-2525. Do you have -- we can be flexible here. Do you want to address -- is there any particular order you'd

in the same transcript, unless you want separate transcripts.

JUDGE CHEN: Panel, any thoughts as to should we --

THE COURT REPORTER: I'm sorry, I had myself muted.

JUDGE JEFFERY: Judge Jeffery here. Yeah, Judge Jeffery here. I think it would be helpful as long as have one transcript -- as long as the three control numbers are identified on the transcript, they should make their way to the respective records, you know, that will make sure that those things get added to each of the records. I don't see the need to, you know, have a new record, but it seems to me that it would automatically find -- as long as the control numbers are all identified on the front face of the transcript, it should make the way to the second record.

JUDGE MCNAMARA: I agree with that. Okay, thank you. All right, let's proceed.

JUDGE CHEN: Okay, hold on. Thumbs up from the court reporter. Okay. All right, so, Mr. Abramson, let's continue to proceed to the next case. Which one would you like to discuss?

MR. ABRAMSON: Whatever the '141 patent is.

JUDGE CHEN: '141. Let's see what number that is.

MR. ABRAMSON: That sounds right. If it's not 3028, then by elimination , that would be the one.

JUDGE CHEN: Okay. All right, let's give me a few minutes to get the 3319 up. Okay. Is the panel ready?

JUDGE JEFFERY: Yes.

JUDGE CHEN: Okay, we shall proceed. We can be flexible and give you about 15 minutes for this one.

MR. ABRAMSON: Sure.

JUDGE CHEN: Mr. Abramson, is that acceptable?

MR. ABRAMSON: Sure.

JUDGE CHEN: Okay, and this deals with the Carmel reference; is that right?

MR. ABRAMSON: Correct.

JUDGE CHEN: Okay, once again, for the record, we are now discussing Appeal Number 2023-002319. Okay, you may begin, Counsel. And you have about 15 minutes.

MR. ABRAMSON: Okay, so the '141 patent and the '011, which we'll discuss after in the next appeal after this one, concern what we refer to as a pull implementation. Instead of preloading an additional load of elements in a transmission buffer, the server creates sequentially identified elements on the server side, and the client can request them as needed to establish and maintain a sufficient playback buffer on its end over a channel that is capable of delivering the initial buffer load of elements and the replenishing elements more rapidly than the playback rate.

The premise is to switch the driving action to the client side and that the server can deliver them faster than they will be needed. Client -- Carmel has been miscast all along as a system based on media slices being successively downloaded by a client by repeated individual client HTTP requests for successive elements by their serial ID. Carmel simply works by a different mechanism, which is that the client specifies a starting slice to the server, and the server pushes successive slices from that starting point, which the client downloads without additional element requests, and it

downloads them just by successively receiving what the server is pushing to it.

As we pointed out in our brief, the ITC, in a recent decision, saw Carmel the same way as Patent Owner does. It's in paragraph — it's in page 7 of our reply. The ITC case that we cite as the Westlaw cite. Patent Owner submits that the ITC was correct in its reading of Carmel. So, with that in mind and without -- I don't need to address every single limitation here. Let me address a couple that stand out.

JUDGE CHEN: Hey, Mr. Abramson, this is Judge Chen again. So, does claim 1 specifically recite a pull architecture? Is that correct, you're characterizing as a pull architecture?

MR. ABRAMSON: Yes.

JUDGE CHEN: So, does claim 1 specifically recite a pull architecture?

MR. ABRAMSON: Yes, it does. I mean, that word doesn't use the word pull, but that's what it's describing as a pull architecture. Yeah, absolutely.

JUDGE CHEN: Okay, can you direct us towards which limitations you're referring to? To the way -- I believe your notation was element -- was the examiner's notation element 1C, as in cat. It reads, "providing a server a program to receive requests from the user system." Are you saying that's the same as pull?

MR. ABRAMSON: Well, in the context of claim 1, yes, that's a pull. This particular language says providing a server program to receive requests, corresponding to specified serial identifiers plural and to send media elements to the server responsive to said request.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventors: Harold Edward Price	Group Art Unit: 3992
Application No. 90/014,833	Examiner: WASSUM, Luke S.
Filed: August 25, 2021	Confirmation No. 1250
Title: STREAMING MEDIA DELIVERY SYSTEM	Patent No. 8,327,011

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR REHEARING

In response to the Decision on Appeal dated November 17, 2023 ("DOA"), Appellant respectfully submits this Request for Rehearing pursuant to 37 CFR § 41.52.

TABLE OF CONTENTS

I. ARGUMENT

In its DOA, the Board affirmed the Examiner on only a single basis, anticipation of claims 1 and 4 under 35 U.S.C. § 102 over Hill (U.S. Pat. No. 6,005,600 (Ex. 10 to the Request)). *See* DOA at 6. To support such a finding of anticipation, Hill must disclose, expressly or inherently, each claim limitation, as arranged in the claims. *See Net MoneyIN, Inc. v. VeriSign, Inc.*, 545 F.3d 1359, 1371 (Fed. Cir. 2008). A claim limitation may be inherently disclosed only if it is necessarily present in Hill and not merely probably or possibly present. *See Guangdong Alison Hi-Tech Co. v. ITC*, 936 F.3d 1353, 1364 (Fed. Cir. 2019).

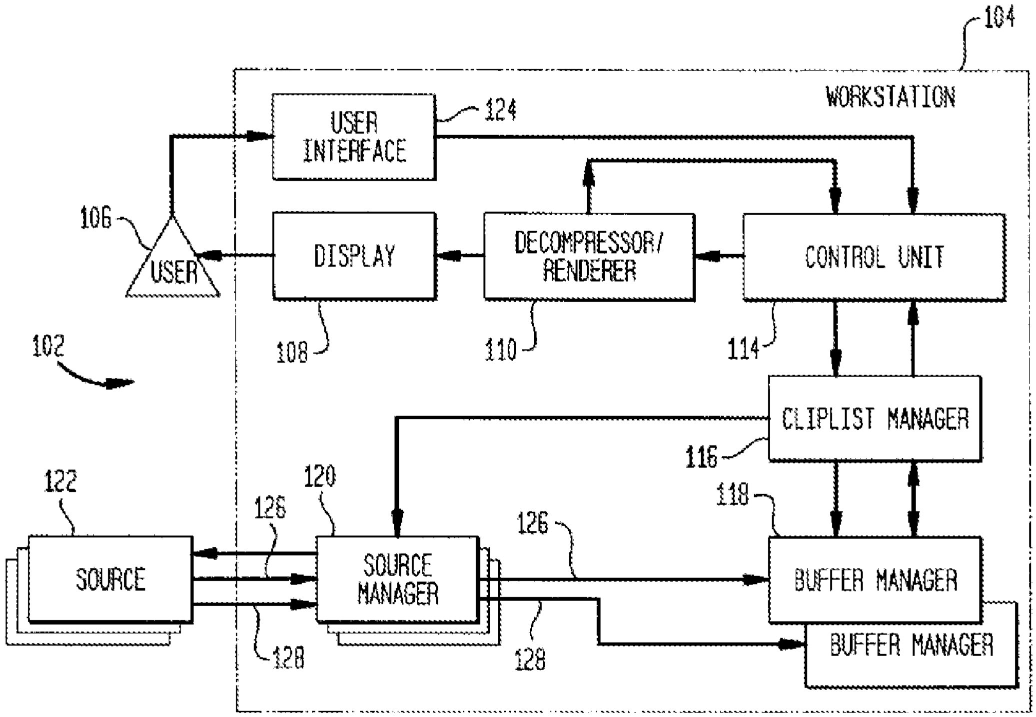
As set forth more fully below, Appellant respectfully submits that the Board made incorrect factual findings concerning Hill and, on that basis, improperly found, *inter alia*, that Hill discloses MIL-2, which recites "transmit to the media source a request to send one or more media data elements, each identified by a serial number." When properly analyzed, Hill does not disclose at least this feature, expressly or inherently, and thus the rejection under 35 U.S.C. § 102 cannot be sustained.[1]

---

[1] The abbreviations and reference labels previously used in Appellant's Appeal Brief are used again in the following arguments herein.

A. Overview of Hill

FIG. 1



Hill, as reflected in Fig. 1 above, concerns a system for viewing a moving picture using a workstation, from "movie data held in various storage devices with varying bandwidth capabilities, where the movie player adapts to these carrying conditions." Hill, 1:7-10. Portions of materials for a movie being worked on in a studio (termed "clips") can reside on different sources, which vary in speed and capabilities. Hill, 2:25-28, 3:46-51. "In typical usage, [the system of Hill] accesses a global playback sequence consisting of several clips." Hill, 11:62-64. For example, a stream currently being edited by an animator may be "stored locally for easy access. In order to develop the scene in context, it is very useful to be able to play the scenes immediately before and after. These surrounding scenes, called 'hook-ups', often are stored as part of a large library accessed via the network." Hill, 2:9-13.

The units of data that Hill's disclosure deals with in obtaining and assembling audio/video content, are "frames." Each video frame has an associated audio frame, and "[a] video frame contains the video information to display a single image." Hill, 3:40-41. Hill refers to example rates of 24 and 48 frames per second. Hill, 7:20-22. Frames have "global" identifiers meaningful to workstation 104 (reflecting their sequence in the entire video being produced on the workstation), as well as "local" identifiers, indicating the frame's location in one of the clips ("local frame number"), together with identification of the clip ("clip number"). *See* Hill, 5:66-6:12.

The clips, which may have various encodings (Hill, 3:51-53), reside with one or more sources 122, which may be local or remotely situated over various types of networks, including, in one example, on a site on the internet. Hill, 3:49-56. Within workstation 104 are one or more "source managers" 120, each for managing a corresponding external source 122. Hill, 4:33-34.

A user of workstation 104 can review the assembled video. *See* Hill, 3:15-18. A buffer is provided in the workstation memory to support such playback. "In order to provide frames quickly, the buffer manager 118 stores a limited number of frames in local digital memory, such as random access memory (RAM) or a local disk drive." Hill, 4:7-11. "The buffer manager 118 then refills its buffers with new frames as they are needed." Hill, 5:12-13. This refill operation, performed internally within workstation 104, is performed as follows:

> the buffer manager 118 sends requests for new frames to the cliplist manager 116. The cliplist manager 116 maintains the mapping between global and local frame numbers. Thus, in step 312, the cliplist manager 116 converts the global frame number to a clip number and local frame number.

> In step 314, the cliplist manager 116 requests the frames from the appropriate source manager 120. The source manager 120 in step 316 retrieves the requested frame data from the source 122.

Hill, 6:7-16.

Each source manager 120 has its own interface with a corresponding external source 122, but Hill *does not disclose* the actual communications between any source manager 120 and its corresponding source 122. The only disclosure in Hill concerning the underlying operations of source managers 120 with respect to sources 122 is:

> The cliplist manager 116 requests frames from a particular source 122 via the appropriate source manager 120. The source manager 120 provides the necessary interface (i.e., hardware or software) for communicating with a source 122. The digital network environment 102 includes an arbitrary number of sources (N_c) from which frames may be accessed. The workstation 104 provides one source manager 120 for each source 122.

Hill, 4:27-34.

Hill provides no further disclosure concerning communications between source managers 120 and their respective sources 122. Disclosure regarding requests by cliplist manager 116 go no further than describing the request that cliplist manager makes to source manager 120, leaving unspecified the details of the referenced "interface" as to how each source manager 120 may structure its communications with its corresponding external source 122.

B. Factual Findings by the Board Concerning Hill

The Board found that "buffer manager 118 of Hill, which retrieves files from source 122, corresponds to the limitation 'transmit to the media

source a request to send one or more media data elements, each identified by a serial number.'" DOA at 6.

In support of this finding, the Board observed that "buffer manager 118 provides frames upon request to the control unit 114 via the cliplist manager 116 (col. 4, 11. 7-8) and the cliplist manager 116 requests frames from a particular source 122 via the appropriate source manager 120 (col. 4, 11. 27-28)." DOA at 7 (internal quotations and alterations omitted). According to the Board:

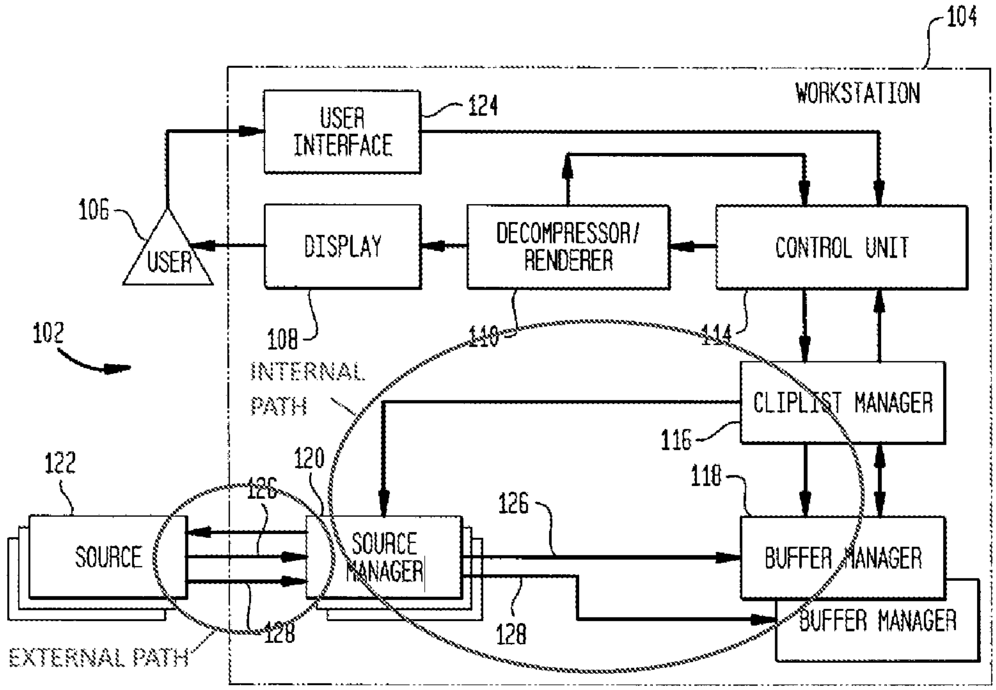
> Because Hill explains that: (i) buffer manager 118 sends requests for new frames to the cliplist manager 116 to retrieve frames from the appropriate source 122; and (ii) the frame request from cliplist manager 116 includes a *global* frame number, Hill describes the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number."

DOA at 9 (emphasis added). The Board reasoned that since Hill "explains that: (i) each source 122 may be accessed either locally or via a network server or a source may be a site on the Internet (col. 3, 11. 53-56); and (ii) the buffer manager 118 receives a frame request from cliplist manager 116, such that the request includes a global frame number (col. 5, 11. 31-33)," that "Hill describes the limitation 'transmit to the media source a request to send one or more media data elements, each identified by a serial number.'" DOA at 10 (internal quotations and alterations omitted).

C. The Board Improperly Maps Internal Communications Within Hill's Workstation onto Communications with External Sources

As discussed above, Hill discloses supplying frames to workstation 104, which frames originate from external sources 122. As disclosed, this

takes place over two distinct path segments, consisting of internal and external portions:



Hill, Fig. 1 (marked-up). As set forth below, there is no explicit or inherent disclosure in Hill of the use of serial identifiers in requests made on the external path.

1. *Hill does not explicitly disclose use of serial identifiers in connection with requests to external sources*

The DOA at 7, applying Hill as a matter of anticipation under § 102, has misidentified the relevant "requests," ascribing these to Hill's buffer manager 118, whereas it is clear from Hill at 4:27-34 that it is cliplist manager 116 that makes the requests, and that it does so "via the appropriate source manager 120," which is a component *internal* to workstation 104.

The DOA fails to point to disclosure of particular requests "each identified by a serial number," as claimed, originating from the workstation 104 and going out over a path external to workstation 104, to media source 122, the latter being the relevant claimed "media source over an Internet

protocol network.” As noted above, Hill goes no further than to state that “the source manager 120 in step 316 retrieves the requested frame data from source 122.” Hill does not say what that request constitutes nor when or how the external retrieval (the one between source manager 120 and the corresponding source 122) is carried out.

The DOA at 10 (first full paragraph) improperly relies on *internal operations* of buffer manager 118 and thus improperly conflates Hill’s *global* frame numbers with a *local* clip frame numbers to find anticipation, notwithstanding the lack of disclosure in Hill that a *global* frame number can even serve as an identifier of data residing on a particular source 122 (indeed, implying that global frame numbers would not be suitable for such identification). In fact, global frame numbers are not meaningful to external sources 122, which only house clips whose content will be *combined* into the global presentation. *E.g.*, Hill, Fig. 7 (local frame numbers each starting at 1). At page 11, the DOA again looks to a request including a “global frame number,” which is not an identification meaningful to external sources 122, and thus cannot correspond to the claimed “requests.”

Although Hill does disclose that “cliplist manager 116 converts the global frame number to a clip number and local frame number” (Hill, 6:10-12), there is no disclosure at all in Hill of any usage of local frame numbers in requests going out along the external path between source managers 120 and the corresponding sources 122, and thus no disclosure in Hill of “transmit[ting] to the media source a request to send one or more media data elements, each identified by a serial number,” in which the “media source [is] over an Internet protocol network,” as recited in claim 1.

2. *Hill does not inherently disclose use of serial identifiers in requests to external sources*

To the extent that inherency in such external communications is relied upon, the Board cites no factual underpinning as to why any undisclosed request in Hill must necessarily include a requested frame number. It is uncontested that Hill is silent on the issue as to how requests for media are made to sources 122, disclosing only that "source manager 120 in step 316 retrieves the requested frame data from the source 122" (Hill, 6:14-16), and nothing more. Given that inherency is a question of fact, without any factual support, any inherency argument must fail. *PAR Pharm., Inc. v. TWI Pharm., Inc.*, 773 F.3d at 1186, 1194 (Fed. Cir. 2014) ("inherent teaching of a prior art reference is a question of fact."); *id.*, at 1195-96 (establishing inherency requires "meet[ing] a high standard" to establish that a "limitation at issue necessarily must be present" in the prior art). Simply by way of one example, Hill's client could request a download of the entire media file or could implement a "request next" protocol like U.S. Patent 7,529,806 to Shteyn (relied upon in Grounds 1 and 4). Merely selecting one alternative out of many does not rise to inherency and thus cannot support alleged invalidity under 35 U.S.C. § 102.

D. The Board Improperly Maps a Record of a Last Global Frame Number Requested with a Record of a Last Local Clip Frame Number Received

ML1-1, which recites "maintain a record of the serial number of the last media data element that has been received," can only be met if Hill explicitly or inherently discloses workstation 104 keeping a record of the *local clip frame number* it last *received*. No such disclosure is present in Hill.

The DOA relies on disclosure in Hill that "buffer manager 118 uses the global frame number last requested as the starting point for determining which frame to request next." DOA at 11. But, as noted above, buffer manager 118 does not communicate with external source 122 at all. Cliplist manager 116 does so, but only indirectly, via source manager 120. All buffer manager 118 knows are the *global* sequence positions of the frames in its buffer (relative to the entire assembled video program) and if any are needed under its "request threshold." Hill, 7:11-15. There is no determinable relationship between that and knowing the serial number (clip number and local frame number) of the last frame that has been received "by the media player," *i.e.*, received by workstation 104 from external source 122, corresponding to what is recited in ML1-1.

The DOA also incorrectly assumes that sequential frames as required will come from the same source 122 as used up to that point, and that the last frame *requested* is necessarily the last one *received*. But vagaries of the internet assure that such is not *necessarily* the case, and thus such a feature cannot be assumed to be present inherently, either.

"Last requested" does not equate to "last received" unless the medium in question can be considered deterministic. There is no disclosure or cited inherent requirement that frames are received in the order of request. The DOA's implicit assumption to this effect is unfounded.

II. CONCLUSION

For the reasons set forth above, Patent Owner respectfully requests rehearing of the DOA and that the panel direct that a reexamination certificate issue for claims 1 and 4 as written.

Dated: January 16, 2024

Respectfully submitted,

LISTON ABRAMSON LLP

/M. Michael Lewis/

M. Michael Lewis (Reg. No. 50,478)
Ronald Abramson (Reg. No. 34,762)
Ari J. Jaffess (Reg. No. 74.558)
405 Lexington Avenue, 46th Floor
New York, NY 10174
(212) 257-1630
Counsel for Patent Owner WAG Acquisition LLP

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte WAG ACQUISITION, LLC

Patent Owner and Appellant

U.S. Pat. No. 8,327,011

Appeal 2023-002525

Ex parte Reexamination Control 90/014,833

PATENT OWNER'S NOTICE OF APPEAL

via email (efileSO@uspto.gov)
Director of the United States Patent and Trademark Office

via EFS and USPS Priority Mail Express®
Patent Trial and Appeal Board
United States Patent and Trademark Office
PO Box 1450
Alexandria, Virginia 22313-1450

via CM/ECF
United States Court of Appeals for the Federal Circuit

Pursuant to 35 U.S.C. §§ 141(b), 142, and 37 C.F.R. §§ 90.2(a), 90.3(a), notice is hereby given that Patent Owner WAG ACQUISITION, LLC, ("Patent Owner") hereby appeals to the United States Court of Appeals for the Federal Circuit from the Decision on Appeal of the Patent Trial and Appeal Board, entered on November 17, 2023, in appeal 2023-002525 from *Ex parte* reexamination control no. 90/014,833 (a copy of which is attached as Appendix A), and from all underlying findings, orders, decisions, rulings, and opinions. This notice is timely filed within 63 days of the April 11, 2024, Decision on Request for Rehearing. 37 C.F.R. § 90.3(b)(1).

Patent Owner indicates that the issues on appeal include, but are not limited to, the Board's determinations with respect to (i) claim construction, (ii) teachings of the asserted art, and (iii) patentability of claims 1 and 4 of U.S. Patent No. 8,327,011 over the asserted art, and (iv) its findings supporting or relating to the aforementioned issues.

A copy of this Notice of Appeal is being filed with the Patent Trial and Appeal Board via mail in accordance with 37 C.F.R. § 90.2(a)(1)(ii) and 37 C.F.R. § 41.10(a), with the Director of the United States Patent and Trademark Office via email in accordance with 37 C.F.R. § 90.2(a)(1)(i), and served upon the reexamination requestor via mail in accordance with 37 C.F.R. § 90.2(a)(3)(i) and 37 C.F.R. § 1.550(f). In addition, this Notice of Appeal and the required fee are

being submitted to the Clerk's Office for the United States Court of Appeals for the Federal Circuit.

Dated: June 13, 2024

By: /M. Michael Lewis/
Mord Michael Lewis
Reg. No. 50,478

By: /Ronald Abramson/
Ronald Abramson

By: /s/ Ronald Abramson
Ronald Abramson

(Lead Counsel)
Reg. No. 34,762
Liston Abramson LLP
405 Lexington Ave., 46th Floor
New York, NY, 10174
Telephone: 212-257-1643
ron.abramson@listonabramson.com

Attorney for Patent Owner

CERTIFICATE OF SERVICE (37 C.F.R. § 1.550(f))

I hereby certify that on this date, I caused a true and correct copy of the foregoing Patent Owner's Notice of Appeal in connection with U.S. Patent No. 8,327,011 to be served via USPS First Class Mail on the following correspondence address of third-party requestor:

Frank M. Gasparo
VENABLE LLP
1270 Avenue of the Americas, 24th Floor
New York, NY 10020
T 212-370-6273
F 212-307-5598
fmgasparo@Venable.com
Attorney for Requesters

Dated: June 13, 2024

Respectfully submitted,

LISTON ABRAMSON LLP
/M. Michael Lewis/
M. Michael Lewis (Reg. No. 50,478)
405 Lexington Avenue, 46th Floor
New York, NY 10174
(212) 257-1630
Counsel for Patent Owner WAG Acquisition LLP

Appendix A

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/014,833	08/25/2021	8327011	125737.538655	1250

7590 11/17/2023

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES, LLC
231 SOMERVILLE ROAD
BEDMINSTER, NJ 07921

EXAMINER	
WASSUM, LUKE S	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
11/17/2023	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte WAG ACQUISITION, LLC
Patent Owner and Appellant

Appeal 2023-002525
Reexamination Control 90/014,833
Patent 8,327,011 B2
Technology Center 3900

Before JOHN A. JEFFERY, ERIC B. CHEN, and BRIAN J. McNAMARA, *Administrative Patent Judges.*

CHEN, *Administrative Patent Judge.*

DECISION ON APPEAL

Pursuant to 35 U.S.C. §§ 134(b) and 306, Patent Owner[1] appeals from the final rejection of claims 1 and 4. Claims 2 and 3 are not subject to reexamination.

A video oral hearing was held on September 18, 2023. The record includes a written transcript of the oral hearing. We have jurisdiction under 35 U.S.C. § 6(b).

We AFFIRM.

---

[1] Patent Owner identifies the real party in interest as WAG Acquisition, LLC. (Appeal Br. 1.)

STATEMENT OF THE CASE

Reexamination Proceedings

A request for *ex parte* reexamination of U.S. Patent No. 8,327,011 B2 ("the '011 patent") was filed on August 25, 2021 and assigned Control No. 90/014,833. The '011 patent, entitled "Streaming Media Buffering System" issued December 4, 2012 to Harold Edward Price, based on Application No. 13/374,942, filed January 24, 2012. The '011 patent is part of a series of multiple continuing applications, with the first application filed on March 28, 2001, which claims priority to provisional Application No. 60/231,997, filed on September 12, 2000.

Claimed Subject Matter

The claims are directed to streaming media, sent via the Internet, such that the media in the user buffer accumulates and immediately played on a user's computer. (Abstract.)

Related Litigation

The '011 patent has been asserted in the following litigation: (i) *WAG Acquisition, LLC v. Sobonito Investments, Ltd. et al.*, No. 2:14-cv-1661 (D.N.J.) (dismissed); (ii) *WAG Acquisition, LLC v. Multi Media, LLC et al.*, No. 2:14-cv-2340 (D.N.J.) (dismissed); (iii) *WAG Acquisition, LLC v. Data Conversions, Inc. et al.*, No. 2:14-cv-2345 (D.N.J.) (dismissed); (iv) *WAG Acquisition, LLC v. Flying Crocodile, Inc. et al.*, No. 2:14-cv-2674 (D.N.J.); (v) *WAG Acquisition, LLC v. Gattyán Group S.á.r.l. et al.*, No. 2:14-cv-2832 (D.N.J.) (dismissed); (vi) *WAG Acquisition, LLC v. FriendFinder Networks Inc., et al.*, No. 2:14-cv-3456 (D.N.J.); (vii) *WAG Acquisition,*

LLC v. Vubeology, Inc. et al., No. 2:14-cv-4531 (D.N.J.) (dismissed); (viii) *WAG Acquisition, LLC v. Gamelink Int'l Ltd. et al.*, No. 2:15-cv-3416 (D.N.J.) (dismissed); and (ix) *WAG Acquisition LLC v. WebPower, Inc. et al.*, No. 2:15-cv-3581 (D.N.J.) (dismissed).

The '011 patent was subject to the following petitions for *inter partes* reviews: (i) *FriendFinder Networks Inc. et al. v. WAG Acquisition LLC*, IPR2015-01033 (PTAB Oct. 19, 2015) (institution denied); (ii) *I.M.L. SLU v. WAG Acquisition LLC*, IPR2016-01655 (PTAB Feb. 27, 2017) (institution denied); and (iii) *WebPower v. WAG Acquisition LLC*, IPR2016-01161 (PTAB Dec. 12, 2016) (institution denied).

The Claims

Claims 1 and 4 are illustrative of the claimed subject matter, and reproduced below with disputed limitations in italics:

> 1. A media player for receiving an audio or video program, the program comprising media data elements, from a media source over an Internet protocol network, and playing the program for a user of the media player, wherein each of the media data elements is associated with a serial number, comprising
>
> a processor;
>
> a memory;
>
> a connection to the network; and
>
> media player software comprising
>
> instructions to cause the media player to request from the media source a predetermined number of media data elements;
>
> instructions to cause the media player to receive media data elements sent to the media player by the

media source and store the media data elements in the memory;

instructions to implement a player buffer manager, for managing a player buffer established in the memory, operable to *maintain a record of the serial number of the last media data element that has been received and stored in the player buffer*;

instructions to cause the media player to play media data elements sequentially from the player buffer; and

instructions to cause the media player to *transmit to the media source a request to send one or more media data elements, each identified by a serial number*, and to repeat transmitting the requests to the media source for sequential media data elements so as to maintain the pre-determined number of media data elements in the player buffer until the last media data element comprising the program has been received.

4. The media player of claim 1, *wherein the instructions* for causing the media player to request from the media source a predetermined number of media data elements *further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player*.

REFERENCES

Name	Reference	Date
Glaser et al.	US 5,793,980	Aug. 11, 1998
Imai et al.	US 5,987,510	Nov. 16, 1999
Hill	US 6,005,600	Dec. 21, 1999
Carmel et al.	US 6,389,473 B1	May 14, 2002
Shteyn	US 7,529,806 B1	May 5, 2009

REJECTIONS

A. Claims 1 and 4 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Shteyn.

B. Claim 4 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Shteyn and Carmel.

C. Claim 4 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Shteyn and Glaser.

D. Claims 1 and 4 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Hill.

E. Claim 4 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Hill and Carmel.

F. Claim 4 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Hill and Glaser.

G. Claims 1 and 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Shteyn and Hill.

H. Claims 1 and 4 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Carmel.

I. Claims 1 and 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Carmel and Hill.

J. Claim 1 stands rejected under 35 U.S.C. § 102(e) as being anticipated by Imai.

K. Claims 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Imai and Carmel.

L. Claims 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Imai and Glaser.

OPINION

§ 102 Rejection—Hill

Independent Claim 1

We are unpersuaded by Patent Owner's arguments (Appeal Br. 34–35; *see also* Reply Br. 22–23) that Hill does not describe the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number," as recited in independent claim 1.

The Examiner found that buffer manager 118 of Hill, which retrieves files from source 122, corresponds to the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number." (Final Act. 15–16; *see also* Ans. 25–26.) We agree with the Examiner's findings.

Hill relates "to playing movie data held in various storage devices with varying bandwidth capabilities, where the movie player adapts to these varying conditions." (Col. 1, ll. 7–10.) Figure 1 of Hill, reproduced below, illustrates digital network environment 102. (Col. 3, ll. 27–28.)

FIG. 1

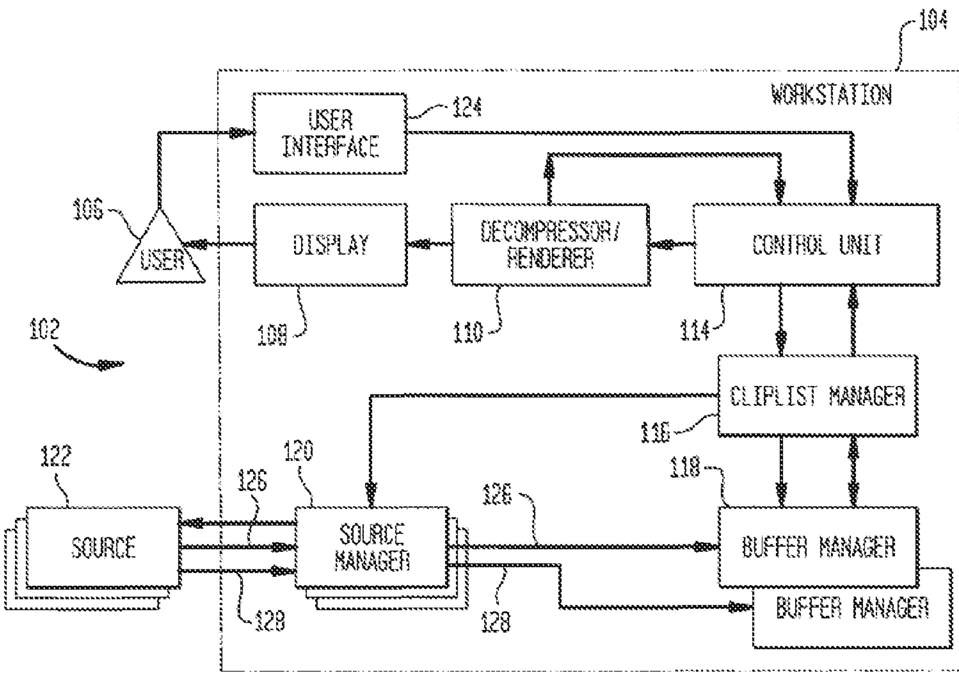


Figure 1 illustrates digital network environment 102.

In reference to Figure 1, Hill explains that digital network environment 102 features "[a] user 106 [that] utilizes the workstation 104 to play movie data." (Col. 3, ll. 27–30.) Hill further explains that "the movie data of interest to the user contains N_c clips, each stored in a separate source 122," such that "[e]ach source 122 may be any type of digital memory, and may be accessed either locally or via a network server" or "a source may be a site on the Internet." (Col. 3, ll. 49–56.) Additionally, Hill also explains that "[t]he buffer manager 118 provides frames upon request to the control unit 114 via the cliplist manager 116" (col. 4, ll. 7–8) and "[t]he cliplist manager 116 requests frames from a particular source 122 via the appropriate source manager 120" (col. 4, ll. 27–28).

Figure 3 of Hill, reproduced below, illustrates flowchart 320 of the input/output operation for buffer manager 18. (Col. 2, ll. 64–65.)

FIG. 3

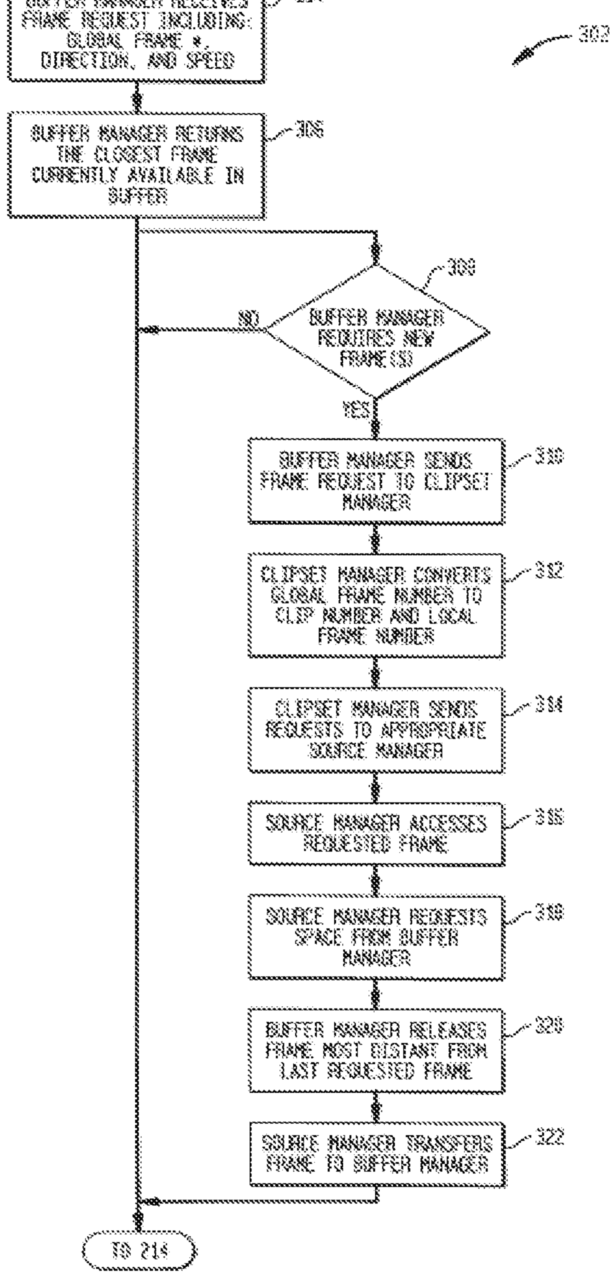


Figure 3 illustrates flowchart 302 for the input/output operation of buffer manager 18.

In reference to Figure 3, Hill illustrates flowchart 302, in which "buffer manager 118 responds to requests for frames from cliplist manager 116 then determines whether new frames need to be loaded" and "[i]f new frames are required, it uses the cliplist manager 116 to retrieve these frames from the appropriate source 122." (Col. 5, ll. 24–28.) In particular, Hill explains that: (i) "[i]n step 304, the buffer manager 118 receives a frame request from cliplist manager 116" such that "[t]he request includes a global frame number" (col. 5, ll. 31–33); (ii) "[i]n step 310, the buffer manager 118 sends requests for new frames to the cliplist manager 116" and "maintains the mapping between global and local frame" (col. 6, ll. 7–10); and (iii) "in step 312, the cliplist manager 116 converts the global frame number to a clip number and local frame number" (col. 6, ll. 10–12).

Because Hill explains that: (i) buffer manager 118 sends requests for new frames to the cliplist manager 116 to retrieve frames from the appropriate source 122; and (ii) the frame request from cliplist manager 116 includes a global frame number, Hill describes the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number."

Patent Owner argues the following:

> [W]hile describing the internal operation of the "source manager" 120 relative to "buffer manager["] "118" at considerable length, Hill does not disclose how source manager 120 (within the player) actually obtains media data from source 122 (outside the player). In a "black box" manner, relative to the actual media source 122, the disclosure states: "The source manager 120 provides the necessary interface (i.e., hardware or software) for communicating with a source 122." "The source manager 120 in step 316 retrieves the requested frame data

> from the source 122." But there is no disclosure of how media data elements are requested from source 122.

(Appeal Br. 34–35 (citations omitted); *see also* Reply Br. 22–23.) Contrary to Patent Owner's arguments that Hall operates in "a 'black box' manner" with "no disclosure of how media data elements are requested from source 122," as discussed previously, Hall explains that: (i) "[e]ach source 122 . . . may be accessed either locally or via a network server" or "a source may be a site on the Internet" (col. 3, ll. 53–56); and (ii) "the buffer manager 118 receives a frame request from cliplist manager 116," such that "[t]he request includes a global frame number" (col. 5, ll. 31–33).

Therefore, we agree with the Examiner that Hill describes the limitation "transmit to the media source a request to send one or more media data elements, each identified by a serial number."

We are further unpersuaded by Patent Owner's arguments (Appeal Br. 35; *see also* Reply Br. 23–24) that Hill does not describe the limitation "maintain a record of the serial number of the last media data element that has been received and stored in the player buffer," as recited in independent claim 1.

The Examiner found that buffer manager 118 of Hill, which uses the last global frame number requested for the next request, corresponds to the limitation "maintain a record of the serial number of the last media data element that has been received." (Final Act. 57; *see also* Ans. 27.) We agree with the Examiner's findings.

As discussed previously, in reference to Figure 3, Hill explains that "[i]n step 304, the buffer manager 118 receives a frame request from cliplist manager 116" and "[t]he request includes a global frame number." (Col. 5,

ll. 31–33.) Moreover, Hill explains that "buffer manager 118 normally uses the global frame number last requested as the starting point for determining which frame to request next." (Col. 10, ll. 4–6.) Because Hill explains that: (i) buffer manager 118 receives a frame request from cliplist manager 116; and (ii) buffer manager 118 uses the global frame number last requested as the starting point for determining which frame to request next, Hill describes the limitation "maintain a record of the serial number of the last media data element that has been received and stored in the player buffer."

Patent Owner argues the following:

> Hill discloses individual frame requests in the particular circumstance where it needs catch up the buffer if it is falling behind in streaming. However, it refers to these particular frames as simply one or two "new" frames. To determine which frame to request in that limited circumstance, Hill discloses "adding one to the last request." There is no teaching in Hill to maintain a record of the serial number of the last received segment.

(Appeal Br. 35 (citation and emphases omitted); *see also* Reply Br. 23–24.) Contrary to Patent Owner's arguments, Hill explains that "buffer manager 118 . . . uses the global frame number last requested as the starting point for determining which frame to request next" (col. 10, ll. 4–6) and, as such, Hill describes the limitation "maintain a record of the serial number of the last media data element that has been received and stored in the player buffer."

Therefore, we agree with the Examiner that Hill describes the limitation "maintain a record of the serial number of the last media data element that has been received and stored in the player buffer."

Thus, we sustain the rejection of independent claim 1 under 35 U.S.C. § 102(e).

Dependent Claim 4

Last, we are unpersuaded by Patent Owner's arguments (Appeal Br. 26, 36; *see also* Reply Br. 24–25) that Hill does not describe the limitation "wherein the instructions . . . further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player," as recited in dependent claim 4.

The Examiner found that Figure 4A of Hill, which includes step 480 for determining buffer fill level, corresponds to the limitation "wherein the instructions . . . further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player." (Final Act. 17–18; *see also* Ans. 28–29.) We agree with the Examiner's findings.

Figure 4A of Hill, reproduced below, illustrates the basic operation of a preferred buffer management strategy. (Col. 2, ll. 66–67.)

FIG. 4A

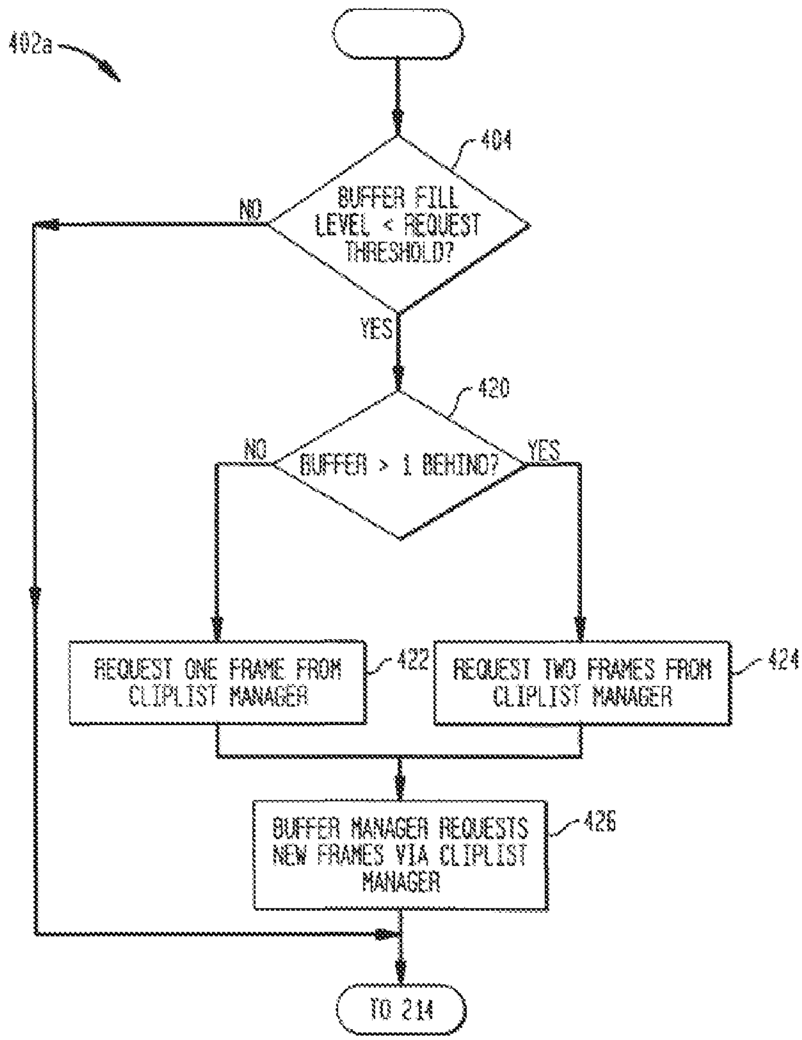


Figure 4A illustrates a flow chart of buffer management strategy.

In reference to Figure 4A, Hill explains that: (i) "the buffer manager 118 determines whether the buffer fill level has dropped below the request threshold, as shown in step 404" (col. 6, ll. 58–60); and (ii) "[i]n step 420, the buffer manager 118 determines by how many frames the request threshold exceeds the fill level" (col. 7, ll. 11–12). With respect to step 420, Hill explains that for a preferred embodiment, "if the play rate is 24 frames

per second, the sources will not be asked to supply more than 48 frames per second." (Col. 7, ll. 17–22.) Because Hill provides for one embodiment in which source 122 supplies 48 frames per second when the play rate is 24 frames per second, Hill describes the limitation "wherein the instructions . . . further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player."

Patent Owner argues the following:

> It is therefore apparent that in the case the buffer is only one frame behind, the request will be for one frame encoded at 24 frames per second. If the request is for two frames, the server will have to supply 48 frames per second—only the latter necessarily being faster than the playback rate. However, there is no teaching that the frames requested one frame at a time should be sent or received any faster than the 24-frame-per-second playback rate. This does not meet the limitation wherein each requested data element is received more rapidly than the playback rate.

(Appeal Br. 36; *see also* Reply Br. 24–25.) Similarly, Patent Owner agues the following:

> The Federal Circuit [in *WAG Acquisition, LLC v. Webpower, Inc.*, 781 F. App'x 1007 (Fed. Cir. 2019)] . . . , addressing a similar limitation in claim 10 of the related 8,122,141 patent, ruled that "the "rate" in claim 10 refers to the rate at which each requested media data element is transmitted from the server to the user computer" and that "[t]he rate limitation in claim 10 therefore refers to the rate at which requested media data elements are sent, not the overall rate at which data is transmitted from the server to the user computer." Thus, the limitation in question was held to be directed to the requested elements, and specifically to "each" requested element, as reflected in the Federal Circuit decision. Claim 4 recites that player instructions cause "the predetermined

> number of media data elements," meaning each of them, to be received at a rate more rapid than the playback rate.

(Appeal Br. 26 (citations omitted).) However, we are not persuaded by Patent Owner's arguments for the same reasons are articulated by another panel of this Board in *WebPower v. WAG Acquisition L.L.C.*, IPR2016-01238 (PTAB July 16, 2020) (Final Written Decision), which addressed the Federal Circuit remand decision, as follows:

> That conclusion [by Patent Owner] relies on an improper importation of an additional limitation into the claim, namely that *all* requested media data elements must be sent by the server at a rate more rapid than the rate at which the streaming media is played back by a user.

(Slip Op. 19 (emphasis in original).)

> That is, Patent Owner appears to read the word "each" in the Federal Circuit's construction as requiring that all media data elements be transmitted faster than the playback rate. As we summarize above, the context in which the federal Circuit arrived at its construction distinguished from our prior construction of "rate" as corresponding to the "overall rate" of transmission from the server to the user computer, such as might be achieved with multiple links over which data elements are sent to the user system. . . . . We discern nothing in the Federal Circuit's decision that compels Patent Owner's implicit additional requirement that *all* media data elements be transmitted faster than the playback rate.

(*Id.* at 20 (emphasis in original).)

Therefore, we agree with the Examiner that Hill describes the limitation "wherein the instructions . . . further causes the media player to receive the predetermined number of media data elements at a rate more rapid than the rate at which the media data elements are to be played out by the media player."

Thus, we sustain the rejection of dependent claim 4 under 35 U.S.C. § 102(e).

Other Pending Rejections

We do not reach the additional rejections of claims 1 and 4 under either: (i) 35 U.S.C. § 102(e) as being anticipated by Shteyn, Carmel, or Imai; and (ii) 35 U.S.C. § 103(a) as being unpatentable over various combinations of Shteyn, Carmel, Glaser, Hill, and Imai. Affirmance of the anticipation-based rejections discussed previously renders it unnecessary to reach the remaining anticipated and obviousness rejections, as all of pending claims have been addressed and found unpatentable. *Cf. In re Gleave*, 560 F.3d 1331, 1338 (Fed. Cir. 2009).

CONCLUSION

The Examiner's decision rejecting claims 1 and 4 under 35 U.S.C. § 102(e) is affirmed.

DECISION SUMMARY

In summary:

Claims Rejected	35 U.S.C. §	Reference(s)/Basis	Affirmed	Reversed
1, 4	102(e)[2]	Shteyn		
4	103(a)[3]	Shteyn, Carmel		
4	103(a)[4]	Shteyn, Glaser		
1, 4	102(e)	Hill	1, 4	

[2] Cumulative rejection.
[3] Cumulative rejection.
[4] Cumulative rejection.

4	103(a)[5]	Hill, Carmel		
4	103(e)[6]	Hill, Glaser		
1, 4	103(a)[7]	Shteyn, Hill		
1, 4	102(e)[8]	Carmel		
1, 4	103(a)[9]	Carmel, Hill		
1	102(e)[10]	Imai		
4	103(a)[11]	Imai, Carmel		
4	103(a)[12]	Imai, Glaser		
Overall Outcome			1, 4	

TIME PERIOD FOR RESPONSE

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

[5] Cumulative rejection.
[6] Cumulative rejection.
[7] Cumulative rejection.
[8] Cumulative rejection.
[9] Cumulative rejection.
[10] Cumulative rejection.
[11] Cumulative rejection.
[12] Cumulative rejection.

PATENT OWNER/APPELLANT:

ERNEST D. BUFF
ERNEST D. BUFF & ASSOCIATES
231 SOMERVILLE ROAD
BEDMINSTER, NJ 07921

THIRD PARTY REQUESTER:

VANABLE LLP
1290 AVENUE OF THE AMERICAS
NEW YORK, NY 10104-3800

CERTIFICATE OF FILING AND SERVICE

I, hereby certify that, on May 21, 2025, I electronically filed the foregoing with the Clerk of Court using the CM/ECF System, which will send notice of such filing to all registered users.

I further certify that, upon acceptance and request from the Court, the required paper copies of the foregoing will be deposited with United Parcel Service for delivery to the Clerk, UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT, 717 Madison Place, N.W., Washington, D.C. 20439.

/s/ Ronald Abramson
Ronald Abramson
Ari J. Jaffess
LISTON ABRAMSON LLP
The Chrysler Building
405 Lexington Avenue, 46th Floor
New York, New York 10174

Counsel for Appellant